République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 Guelma

جامعة 8 ماي 1945 قالمة
UNIVERSITE 8 MAI 1945 GUELMA

Faculté des mathématiques et informatiques et sciences de la matière
Département de Mathématiques
Laboratoire de domiciliation : Laboratoire de Mathématiques Appliquées et de Modélisation

THÈSE

En Vue de l'Obtention du Diplôme de Doctorat en Troisième Cycle

Domaine : Mathématiques et Informatiques. Filière : Mathématiques Appliquées
Spécialité : Modélisation, contrôle et optimisation
Présentée par

HAMEL Naima

Intitulée

L'interrelation entre les algorithmes de gradient conjugué et ceux de Newton

Soutenue le : 17/02/2024 Devant le Jury composé de :

Nom et Prénom	Grade		
Mr AISSAOUI Mohamed Zine	Prof	Univ. de Guelma	Président
Mr BENRABIA Noureddine	MCA	Univ. de Souk Ahras	Encadreur
Mr GHIAT Mourad	MCA	Univ. de Guelma	Co-encadreur
Mr MESBAHI Salim	Prof	Univ. de Sétif 1	Examinateur
Mme LASKRI Yamina	Prof	ENSTI Annaba	Examinateur
Mr ELLAGGOUNE Fateh	Prof	Univ. de Guelma	Examinateur

Année Universitaire : 2023/2024

A mes parents

Remerciement

J'adresse mes remerciements et ma profonde gratitude à mon directeur de thèse Dr **Noureddine BENRABIA** pour la confiance qu' il m'a accordé, ses encouragements, et ses conseils précieux.

Aussi, Je tiens à remercier mon co-encadrant Dr **Mourad GHAIT** pour ses encouragements.

Je tiens à exprimer mes sincères remerciements et ma reconnaissance au Professeur **Hamza GUEBBAI** pour ses conseils, orientations et précieuses aides tout au long de ces années, et pour son temps précieux pour expliquer toute demande ou question malgré ses obligations et responsabilités.

Toute ma profonde gratitude au Professeur **M. Z. AISSAOUI** le président du jury pour ses orientations, son temps précieux et les judicieux conseils.

J'adresse également mes sincères remerciements aux membres du jury: Pr. **LASKRI Yamina**, Pr. **MESBAHI Salim**, Pr. **ELLAGGOUNE Fateh** qui m'ont fait l'honneur de participer à l'évaluation de ma thèse et de partager avec moi leurs remarques pertinentes et objectives, ce qui a contribué à l'enrichissement de cette recherche.

Un grand merci à l'équipe du **Laboratoire de Mathématiques Appliqué et Modélisation**, j'ai l'honneur d'avoir été membre de ce laboratoire.

Enfin, je n'oublie pas de remercier tous ceux qui ont facilité ma tâche tout au long de ce parcours.

Abstract

The conjugate gradient and Newton's algorithms are highly reliable, with well-established theoretical results and excellent numerical performance. However, their relationship has not been fully explored, and there has been limited progress in designing efficient and robust algorithms for large-scale problems that utilize the concepts of these two algorithm classes. Therefore, the objective of this thesis is to address the issue of the interrelation between the conjugate gradient and Newton's algorithms. To provide at least a partial answer to this problem, we investigate how these two algorithms can influence and complement each other to enhance optimization performance. To achieve this, we develop three conjugate gradient methods based on the direction of Newton and the secant equation. The aim is to preserve the advantages of the conjugate gradient algorithms while incorporating certain beneficial properties of Newton's method without directly evaluating the Hessian matrix. We also study the global convergence of our methods and demonstrate their effectiveness by numerically comparing them to other efficient methods. Additionally, we devise a numerical test to evaluate whether our three algorithms can approximate the quadratic convergence observed in Newton's method.

Keywords: Conjugate gradient algorithm, Newton's algorithm, Descent direction, Global convergence, Quadratic convergence, Numerical experiments.

Mathematics Subject Classification: 90C06, 90C26, 49M15, 90C30, 65K05

Résumé

Les algorithmes du gradient conjugué et de Newton sont très fiables, avec des résultats théoriques bien établis et une excellente expérience numérique. Cependant, la relation entre eux n'a pas été pleinement explorées, et il n'y a pas eu de progrès significatif dans la conception d'algorithmes efficaces et robustes pour des problèmes à grande taille, utilisant les concepts de ces deux classes d'algorithmes. Ainsi, l'objectif de cette thèse est de traiter la problématique " l' interrelation entre les algorithmes du gradient conjugué et ceux de Newton " Afin d'y apporter une réponse, même partielle, nous étudions comment ces deux algorithmes peuvent s'influencer et se compléter mutuellement pour améliorer les performances d'optimisation. Pour ce faire, nous développons trois méthodes du gradient conjugué en se basant sur la direction de Newton et l'équation de la sécante. L'idée est de conserver les avantages des algorithmes de gradient conjugué, tout en incorporant certaines propriétés bénéfiques de la méthode de Newton, sans l'évaluation directe de la matrice hessienne. Nous étudions également la convergence globale de nos méthodes et nous montrons leur efficacité, en les comparant numériquement à d'autres méthodes efficaces. De plus, Nous développons un test numérique pour évaluer si nos trois algorithmes approche numériquement la convergence quadratique observé dans la méthode de Newton.

Mots clé : Algorithme du gradient conjugué, Algorithme du Newton, Direction de descente, Convergence globale, Convergence quadratique, Expériences numériques

Mathematics Subject Classification : 90C06, 90C26, 49M15, 90C30, 65K05

الملخص

تعتبر خوارزميات التدرج المترافق وخوارزميات نيوتن موثوقة للغاية، مع نتائج نظرية راسخة وخبرة عددية ممتازة. ومع ذلك، لم يتم استكشاف العلاقة بينهما بشكل كامل، ولم يكن هناك تقدم كبير في تصميم خوارزميات فعالة وقوية للمشكلات واسعة النطاق، باستخدام مفاهيم هاتين الفئتين من الخوارزميات. وبالتالي فإن الهدف من هذه الأطروحة هو معالجة مشكلة "العلاقة المتبادلة بين خوارزميات التدرج المترافقة وخوارزميات نيوتن" ومن أجل تقديم إجابة ولو جزئية، قمنا بدراسة كيف يمكن أن تؤثر هاتين الخوارزميتين وتكمل كل منهما الأخرى تحسين الأداء الأمثل. للقيام بذلك، قمنا بتطوير ثلاث طرق للتدرج المترافق بناءً على اتجاه نيوتن والمعادلة القاطعة. تتمثل الفكرة في الاحتفاظ بمزايا خوارزميات التدرج المترافق، مع دمج بعض الخصائص المفيدة لطريقة نيوتن، دون التقييم المباشر للمصفوفة الهاسيانة. كما نقوم بدراسة التقارب الشامل لطرقنا وإظهار فعاليتها من خلال مقارنتها عدديا بالطرق الفعالة الأخرى. علاوة على ذلك، قمنا بتطوير اختبار عددي لتقييم ما إذا كانت خوارزمياتنا الثلاثة تقترب عدديًا من التقارب التربيعي الذي لوحظ في طريقة نيوتن.

الكلمات المفتاحية : خوارزمية التدرج المترافق، خوارزمية نيوتن، اتجاه النزول، التقارب العالمي، التقارب التربيعي، التجارب العددية.

تصنيف مواضيع الرياضيات : 90C06, 90C26, 49M15, 90C30, 65K05.

Table des matières

Introduction générale

L'optimisation est une discipline fondamentale des mathématiques appliquée dans les sciences de l'ingénierie, l'économie et l'industrie en général. Son objectif est de trouver la solution optimale à certains problèmes mathématiquement définis, qui sont souvent des modèles de phénomènes réels.

Cette recherche se concentre principalement sur l'étude et le test de quelques problèmes d'optimisation sans contrainte, où les variables sont des nombres réels et les fonctions objectif sont réelles et dépendent de plusieurs variables. Plus formellement, l'optimisation sans contrainte correspond à l'étude de problèmes exprimés de la manière suivante.

Trouver un élément $x^* \in R^n$ qui minimise ou maximise une fonction $f\colon\ R^n \mapsto R$ donnée. En général, on se concentre principalement sur la minimisation des problèmes, car la maximisation et la minimisation sont équivalentes.

Au cours des dernières années, de nombreuses techniques d'optimisation ont été développées pour résoudre les problèmes d'optimisation sans contrainte. Parmi ces techniques les algorithmes de gradient conjugués et de Newton ont été largement étudiés et appliquées avec succès dans de nombreux secteurs. L'algorithme de gradient conjugué a été découverte en 1952 par Hestenes et Steifel [31] (méthode HS) pour minimiser les fonctions quadratiques strictement convexes (cas linéaire). En 1964, Fletcher et Reeves [23] (méthode FR) ont été les premiers à étendre cette méthode pour le cas non linéaire, suivis par plusieurs autres mathématiciens notament, Polak-Ribière et Ployak [45, 46] en 1969 (méthode PRP) , Fletcher en 1987 [24] (méthode CD), Liu et Storey en 1991 [40] (méthode LS), Dai et Yuan en 1999 [15] (méthode DY).
L'algorithme de Newton peut être considéré comme la méthode locale de base utilisant des informations du second ordre. Cette méthode est brièvement illustrée car la plupart des méthodes du second ordre sont dérivées sous forme d'approximations. Lorsque le point initial est proche de la solution, la méthode de Newton fournit un taux de convergence quadratique. Pour plus de détails, voir [49].

Ces deux algorithmes sont réputés pour leur fiabilité, leurs résultats théoriques bien établis et leur excellente performance numérique. Cependant, la relation et l'interconnexion entre eux n'ont pas été pleinement explorées, et il n'y a pas eu de progrès significatif dans la conception d'algorithmes efficaces utilisant les concepts de ces deux catégories d'algorithmes.

Nous pouvons formuler la problématique dans la question suivante "existe-t-il une interrelation entre les algorithmes du gradient conjugué et ceux de Newton?"
Afin de répondre à cette problématique, nous approfondirons notre compréhension de l'interrelation entre les algorithmes du gradient conjugué et de Newton, en étudiant comment ils peuvent s'influencer et se compléter mutuellement pour améliorer les performances des méthodes d'optimisation. Pour ce faire, nous développerons des nouveaux méthodes d'optimisation qui combinent à la fois les avantages des méthodes du gradient conjugué et de Newton, tout en évitant leurs faiblesses.

Notre thèse est structurée en cinq chapitres, précédés d'une introduction et conclue par une conclusion avec des perspectives pour des études futures.

Dans le premier chapitre, nous introduisons un rappel de quelques notions mathématiques importantes relatives à l'optimisation sans contrainte. De plus, nous nous rappellerons les méthodes du gradient conjugué et la méthode de Newton, ainsi que leurs propriétés de convergence.

Dans le deuxième chapitre, qui représente l'essentiel de notre recherche, nous avons développé une nouvelle méthode hybride de gradient conjugué qui combine les méthodes (DY) et (DL) de manière convexe. Cette méthode a été conçue pour être étroitement liée à la méthode de Newton, tout en évitant le coût de calcul associé à l'évaluation directe de la matrice hessienne grâce à l'utilisation de l'équation de sécante. Cela la rend utile pour résoudre des problèmes d'optimisation à grande échelle. La condition de descente suffisante et la convergence globale ont été prouvées.

Nous avons également réalisé des expériences numériques pour confirmer l'efficacité de la nouvelle méthode par rapport à quelques autres méthodes, en termes de temps de calcul, de précision et de nombre d'itérations.

Le troisième chapitre, est consacré à la présentation d'une nouvelle méthode hybride de gradient conjugué qui relie les méthodes (WYL) et (CD) dans un mélange convexe en utilisant la direction de Newton et l'équation de la sécante afin d'obtenir certaines propriétés souhaitables à la fois des méthodes du gradient conjugué et de Newton, tout en évitant leurs inconvénients. La condition de descente suffisante et la convergence globale ont été prouvées. L'efficacité de la méthode proposée a été confirmée à l'aide d'un ensemble de problèmes de test standard, démontrant sa supériorité par rapport aux algorithmes (WYL) et (CD) dans la plupart des cas.

Le quatrième chapitre est consacré à la présentation d'un nouvel algorithme hybride de gradient conjugué. Cet algorithme introduit une approche permettant de combiner trois directions de descente de manière quasi-convexe. L'objectif de cet algorithme est de construire une direction qui approxime étroitement la direction de Newton en utilisant l'équation de la sécante. Nous avons démontré la propriété de descente et la convergence globale de cet algorithme. Nous avons illustré son efficacité en le comparant aux algorithmes DY, HS et DL à partir desquels il a été construit. De plus, nous l'avons également comparé à l'algorithme BFGS pour confirmer que notre nouvel algorithme est une meilleure approximation de la méthode de Newton que l'algorithme BFGS.

Enfin, sachant que, nos trois algorithmes de gradient conjugué (DYDL, WYLCD, QCC) ont été conçue pour être étroitement liée à la méthode de Newton afin de préserver certaines bonnes propriétés des algorithmes de gradient conjugué tout en incorporant certaines propriétés souhaitables de la méthode de Newton. Nous avons donc développé dans le chapitre 5, un nouveau test numérique pour tester si notre trois algorithmes approche numériquement la convergence quadratique observé dans la méthode de Newton, en analysant les rapports d'erreur entre deux itérations successives, et montrant graphiquement comment ces rapports d'erreur indiquent la présence d'un comportement de convergence quadratique.

Chapitre 1

Optimisation sans contraintes

Introduction

L'optimisation sans contrainte est un domaine de l'optimisation, qui vise à trouver le minimum (ou maximum) d'une fonction objectif, sans imposer de restrictions sur les variables. Cela permet de résoudre un un grand nombre de problèmes dans divers domaines, en utilisant des algorithmes spécifiques pour atteindre l'optimum souhaité, notamment les méthodes de gradient conjugué et de Newton.

1.1 Rappel sur quelques notions mathématiques importantes

Dans ce qui suit on va introduire quelques définitions fondamentaux dont on aura besoin dans le traitement de notre sujet.

Soit $\mathbb{R}^n$ l'espace vectoriel réel de dimension $n \in \mathbb{N}$, dans tous ce qui suit, les vecteurs sont toujours des vecteurs colonnes et la transposée de x est notée par x^T.
Nous utilisons la norme $\| . \|$ pour représenter la norme euclidienne, ainsi, la norme de $x \in \mathbb{R}^n$ est donnée par $\| x \|_2 = \sqrt{x^T x}$.

Définition 1. *[13].*
Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable

- *La fonction noté $\nabla f : \mathbb{R}^n \mapsto \mathbb{R}$ est appelé le gradient de f et on écrit le gradient de f au point $x \in \mathbb{R}^n$ comme suit*
$$\nabla f(x) = (\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n}) \tag{1.1}$$
- *Soit $\nabla f : \mathbb{R}^n \mapsto \mathbb{R}$ différentiable, on appelle la Hessienne de f, la fonction $\nabla^2 f : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ définie comme suit*
$$\nabla^2 f(x) = \frac{\partial^2 f}{\partial x_i \partial x_j}(x), \quad i = 1, \dots n, j = 1, \dots, n. \tag{1.2}$$

Définition 2. *.[13]*

- *Une matrice carré $B \in R^{n \times n}$ est dite symétrique si*
$$B = B^T$$
- *Une matrice symétrique $B \in R^{n \times n}$ est dite définie positive si*
$$x^T Bx > 0, \forall x \in \mathbb{R}^n, x \neq 0$$
- *Une matrice symétrique $B \in R^{n \times n}$ est dite semi définie positive si*
$$x^T Bx \geq 0, \forall x \in \mathbb{R}^n$$

L'optimisation sans contrainte consiste à rechercher le minimum d'une fonction f de n variables à valeurs réelles sans imposer de contraintes spécifiques sur ces variables.
Nous cherchons à trouver un point x^* qui vérifié la relation suivante

$$f(x^*) \leq f(x),$$

où, x^* est appelé minimum local.
Ce problème sera formulé de la façon suivante

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1.3}$$

Les méthodes de résolution de ce type de problème sont de nature itérative, c'est à dire qu' à partir d'un point initial x_0, on construit une suite qui converge vers une solution dite optimale.

1.2 Les conditions d'optimalité

La résolution efficace d'un problème d'optimisation, nécessitent des conditions d'optimalité adéquates. En effet, ces conditions nous permettent non seulement de vérifier la validité des solutions obtenues, mais elles jouent également un rôle clé dans le développement des algorithmes d'optimisation.
Considérons le problème de minimisation (1.3) ($\min_{x \in \mathbb{R}^n} f(x)$.)

Définition 3. *. [13]*

1. *$x^* \in \mathbb{R}^n$ est un minimum local de (1.3), s'il existe un voisinage $V_\epsilon(x^*)$ tel que*

$$f(x^*) \leq f(x), \forall x \in V_\epsilon(x^*) \tag{1.4}$$

2. *Dans le cas ou $f(x^*) < f(x)$, Le minimum local est dit strict.*

3. *$x^* \in \mathbb{R}^n$ est un minimum globale de (1.3) si*

$$f(x^*) \leq f(x), \forall x \in \mathbb{R}^n \tag{1.5}$$

Définition 4. *.[13]*
soit la fonction $f : \mathbb{R}^n \mapsto \mathbb{R}$ différentiable en $x \in \mathbb{R}^n$.

1. *Un vecteur $d \in \mathbb{R}^n$ est appelé une direction de descente pour f en x si la condition*

$$\nabla f(x)^T d < 0 \tag{1.6}$$

est vérifié

2. *Le vecteur $d \in \mathbb{R}^n$ est dite une direction de descente suffisante de f en x si*

$$g_k^T d_k \leq c \parallel g_k \parallel^2, \qquad \forall k \in N, \quad c > 0, \tag{1.7}$$

avec $g_k = \nabla f(x_k)$.

Le résultat ci-dessous nous permet de fournir des critères pour déterminer si un point donné est un minimum local ou global de la fonction f. La propriété de différentiabilité de f permet de caractériser une solution optimale. Les théorèmes suivant illustrent *"les conditions nécessaires d'optimalité."*

Théoreme 1 (Optimalité du premier ordre)**.** *.[44]*
Supposons que la fonction $f : \mathbb{R}^n \mapsto \mathbb{R}$ est différentiable au point $x^* \in \mathbb{R}^n$. Si le point x^* est un minimum local de f, alors $\nabla f(x^*) = 0$.

Théoreme 2 (Optimalité du seconde ordre)**.** *.[44]*
Soit la fonction $f : \mathbb{R}^n \mapsto \mathbb{R}$ deux fois différentiable au point $x^* \in \mathbb{R}^n$. Si x^* est un minimum local de f, alors $\nabla f(x^*) = 0$ de plus $x^T \nabla^2 f(x^*) x \geq 0$, $\forall x \in \mathbb{R}^n$ c-à-d $\nabla^2 f(x^*)$ est semi définie positive.

Les conditions précédemment énoncées sont nécessaires, c'est-à-dire qu'elles doivent être vérifiées par tout minimum local. Cependant, il est important de noter que tout point qui satisfait ces conditions n'est pas nécessairement un minimum local. Les théorèmes suivants établis *"les conditions suffisantes"* pour que un point soit un minimum local, si f est deux fois différentiable.

Théoreme 3 (Optimalité du premier ordre). *.[44]*
Soit la fonction $f : \mathbb{R}^n \mapsto \mathbb{R}$ *convexe, supposons que* f *est différentiable en* $x^* \in \mathbb{R}^n$*.*
Si $\nabla f(x^*) = 0$ *alors* x^* *est un minimum globale de* f*.*

Théoreme 4 (Optimalité du seconde ordre). *[44].*
Soit la fonction $f : \mathbb{R}^n \mapsto \mathbb{R}$ *deux fois différentiable en* $x^* \in \mathbb{R}^n$*.*
Si $\nabla f(x^*) = 0$ *et* $\nabla^2 f(x^*)$ *est définie positive alors* x^* *est un minimum local strict de* f*.*

Remarque 1. *.[44]*

1. *Si la fonction* f *est convexe, alors tout minimum local est également un minimum global.*
2. *si la fonction* f *est strictement convexe, alors tout minimum local est un minimum global unique.*

Nous allons maintenant présenté quelques définitions sur les modes de convergence

Définition 5. *.[49]*
Soit $\{x_k\}_{k\in\mathbb{N}}$ *une suite de vecteurs dans* $\mathbb{R}^n$*, qui converge vers* x^**. La convergence est dite,*

- *Linéaire, s' il existe* $l \in]0, 1[$*, tel que*

$$\lim_{k\mapsto\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|} = l$$

- *super linéaire, si*

$$\lim_{k\mapsto\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|} = 0$$

- *super linéaire d'ordre* α*, si*

$$\lim_{k\mapsto\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|^{\alpha}} < \infty, \quad \alpha > 1$$

  En particulier si $\alpha = 2$ *la convergence de* $\{x_k\}_{k\in\mathbb{N}}$ *vers* x^* *est dite quadratique (super linéaire d'ordre 2)*

1.3 La recherche linéaire

En optimisation mathématique, la recherche linéaire est l'une des méthodes classiques employées, pour traiter la convergence des algorithmes de calcul du minimum x^* d'une fonction f: $\mathbb{R}^n \mapsto \mathbb{R}$.

Considérons le problème d'optimisation sans contrainte (1.3),

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.8}$$

pour résoudre le problème (1.3), on fait appel aux algorithmes basés sur la génération de la séquence

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k \geq 0, \tag{1.9}$$

où,

- d_k est la direction de descente.
- α_k est le pas, qui est choisie de manière optimale et qui doit vérifier

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k), \qquad \alpha > 0. \tag{1.10}$$

En d'autres termes, le pas α_k est une solution optimale du problème d'optimisation unidimensionnel suivant

$$\min_{\alpha>0} f(x_k + \alpha d_k), \tag{1.11}$$

où, x_k et d_k sont fixés et la fonction à minimiser notée ϕ_k, est définie comme suit,

$$\phi_k : \mathbb{R}^n \mapsto \mathbb{R}, \quad \alpha \mapsto \phi_k(\alpha) = f(x_k + \alpha d_k). \tag{1.12}$$

Rappelons que si f est différentiable, le pas optimal α_k peut être caractérisé par

$$\begin{cases} \phi_k'(\alpha_k) = 0, \\ \phi_k(\alpha_k) \le \phi_k(\alpha), \quad \text{pour} \quad 0 \le \alpha \le \alpha_k, \end{cases} \tag{1.13}$$

autrement dit, α_k est un minimum local de ϕ_k qui assure la décroissance de f.

Nous allons maintenant présenter les différentes approches utilisées pour déterminer un pas α_k le long d'une direction de descente d_k, ce processus étant communément appelé recherche linéaire. Il existe deux principales catégories de techniques qui se concentrent sur l'optimisation unidimensionnelle, à savoir

1. les recherches linéaires exactes.
2. les recherches linéaires inexactes.

1.3.1 La recherche linéaire exacte

Puisque nous cherchons à minimiser la fonction f, il est naturel de chercher à minimiser le critère le long de la direction de descente d_k, Ainsi, nous devons déterminer la valeur du pas $\alpha_k > 0$ qui résout le problème,

$$\min_{\alpha > 0} \phi_k(\alpha). \tag{1.14}$$

Cette procédure est appelée règle de Cauchy et la taille du pas donnée par cette règle est appelée "pas de Cauchy" ou "pas optimal. Dans certains cas, nous préférons le plus petit point stationnaire de ϕ_k, qui le rende décroissant, nous aurons alors à déterminer α_k comme solution exacte de

$$\alpha_k = \min\{\alpha > 0 : \phi_k'(\alpha) = 0, \quad \phi_k(\alpha) < \phi_k(0)\}$$

comme sera déterminer de maniéré exacte alors cette technique est appelée, la recherche linéaire exacte [49].

Remarque 2. *.[49]*

1. *Dans la plupart des algorithmes d'optimisation modernes la recherche linéaire exacte n'est généralement pas utilisée, car trouver la valeur α_k signifie qu'il va falloir évaluer la fonction ϕ_k à plusieurs reprises, ce qui peut être coûteux en termes de temps de calcul. En pratique, on cherche à trouver une valeur de x_* qui garantit une décroissance suffisante de la fonction f.*
2. *La règle de recherche linéaire exacte est uniquement utilisées dans des cas particuliers, par exemple si la fonction ϕ_k est quadratique, alors la solution de la recherche linéaire exacte s'obtient d'une façon exacte et par un nombre fini d'itérations.*

1.3.2 La recherche linéaire inexacte

Au lieu d'exiger que α_k minimise ϕ_k, il est préférable d'imposer des conditions moins restrictives mais plus facilement vérifiables et contribuent toujours à la convergence des algorithmes. En adoptant cette approche, il ne s'agit plus de trouver un unique pas (ou quelques pas) qui satisfait ces conditions, mais plutôt de définir un intervalle de pas (ou plusieurs intervalles), ce qui facilite la recherche d'une valeur approchée du pas α_k qui assure la convergence. C'est ce que nous faisons avec les règles d'Armijo, Goldstein, Wolfe et Wolfe forte.

Règle d'Armijo [10] La Règle d'Armijo est basée sur le choix d'un paramètre ρ, avec

$$0 < \rho < 1$$

consiste à déterminer une valeur approchée de α_k par cette condition

$$\phi_k(\alpha) \le \phi_k(0) + \alpha\rho\phi_k'(0). \tag{1.15}$$

Cette règle a l'inconvénient de favoriser les valeurs trop petites. Elle n'est pas souvent utilisé seul.

Remarque 3. *.*
Accepter des pas trop petits peut entrainer un risque de fausse convergence.

Règle de Goldstein et Price [25]
La règle d'Armijo permet de garantir la décroissance de la fonction f à chaque pas, mais cela n'est pas toujours suffisant pour assurer la convergence. La règle de Goldstein et Price suffit à garantir la convergence sous certaines conditions, elle détermine les valeurs approchées de α_k à l'aide de deux conditions suivantes:

$$\begin{cases} \phi_k(\alpha) \leq \phi_k(0) + \alpha\rho\phi_k^{'}(0), \\ \phi_k(\alpha) \geq \phi_k(0) + \alpha\delta\phi_k^{'}(0) \end{cases} \tag{1.16}$$

avec $0 < \rho < \delta < 1$, la deuxième inégalité qui empêche le pas d'être trop petit.
Règle de Wolfe [51]
La règle de Goldstein et Price peuvent exclure un minimum, ce qui est peut être un inconvénient. La règle de Wolfe remédie à cet inconvénient. En effet, étant donnés deux paramètres ρ, δ tel que $0 < \rho < \delta < 1$

$$\begin{cases} \phi_k(\alpha) \leq \phi_k(0) + \alpha\rho\phi_k^{'}(0), \\ \phi_k^{'}(\alpha) \geq \delta\phi_k^{'}(0) \end{cases} \tag{1.17}$$

Remarque 4. *La règle de Wolfe nécessite le calcul de $\phi_k^{'}$, elle est donc théoriquement plus couteuse que la règle de Goldstein et Price. Cependant, dans de nombreuses applications, le calcul du gradient $\nabla f(x)$ représente un petit cout supplémentaire par rapport au cout de l'évaluation de $f(x)$, c'est pourquoi cette règle est largement utilisée.*

Règle de Wolfe forte [52]
Pour certains méthodes (par exemple la méthode de gradient conjugué non linéaire), il peut être nécessaire d'imposer une condition plus restrictive que celle donnée par la deuxième inégalité dans (1.17). Par conséquent, la deuxième inégalité dans (1.17) est remplacée par,

$$\begin{cases} \phi_k(\alpha) \leq \phi_k(0) + \alpha\rho\phi_k^{'}(0), \\ |\phi_k^{'}(\alpha)| \leq -\delta\phi_k^{'}(0) \end{cases} \tag{1.18}$$

où, $0 < \rho < \delta < 1$.

1.4 La méthode du gradient conjugué

1.4.1 La méthode du gradient conjugué linéaire

L'algorithme du gradient conjugué linéaire, sert à minimiser des fonctions quadratiques convexes, ou pour résoudre des systèmes d'équations linéaires avec des matrices définies positives. Cet algorithme a été présenté pour la première fois par Hestenes et Stiefel [31] en (1952).

Soit $G \in \mathbb{R}^{n.n}$ une matrice symétrique définie positive et un vecteur $b \in \mathbb{R}^n$. Considérons le problème de minimisation d'une fonction quadratique f à savoir

$$\min\{f(x) : x \in \mathbb{R}^n\} = \min\{\frac{1}{2}x^TGx - b^Tx : x \in \mathbb{R}^n\} \tag{1.19}$$

nous avons

$$g_k = \nabla f(x) = Gx - b \qquad \nabla^2 f(x) = G. \tag{1.20}$$

Lorsque G est définie positive, la fonction f est strictement convexe, et selon les conditions d'optimalité pour la minimisation d'une fonction différentiable, il s'ensuit que la fonction f admet un minimum global unique x^* vérifiant $\nabla f(x^*) = 0$.

Notons que, d'après (1.20), x^* la solution du problème (1.19) est également une solution du système linéaire suivant:

$$Gx = b$$

Si on note $\nabla f(x_k) = g_k$, l'algorithme de la méthode du gradient conjugué dans le cas des fonctions quadratiques, génère une séquence comme suit,

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k, \tag{1.21}$$

où,

- d_k est la direction de recherche.
- α_k est le pas de recherche déterminé par une recherche linéaire exacte/inexacte.

Considérons le vecteur d_0 fixé, les directions $d_1, d_2, \ldots, d_{n-1}$ sont calculées à chaque itération par la relation,

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 0, \tag{1.22}$$

où, β_k est une suite des coefficients obtenu de sorte que d_k soit G conjugué avec les autres vecteurs, c'est à dire le choix des coefficients β_k permet d'assurer la propriété suivante:

$$d_{k+1}^T G d_k = 0, \tag{1.23}$$

par conséquent, on en déduit

$$d_{k+1}^T G d_k = 0$$

En remplaçant la formule de la direction d_{k+1} par 1.22, nous obtenons

$$(-g_{k+1} + \beta_k d_k)^T G d_k = 0$$

on a alors

$$\beta_k d_k^T G d_k = g_{k+1}^T G d_k$$

Donc

$$\beta_k = \frac{g_{k+1}^T G d_k}{d_k^T G d_k}, \tag{1.24}$$

ainsi, le coefficient β_k est bien défini.

Dans le cas d'une fonction quadratique f, on peut facilement déterminer le pas α_k en tant que solution analytique exacte du problème de minimisation unidimensionnel,

$$\alpha_k = \min_{\alpha > 0} f(x_k + \alpha d_k). \tag{1.25}$$

En d'autres termes le pas optimal α_k vérifié $f^{'}(\alpha_k) = 0$. Or

$$f^{'}(\alpha_k) \quad = \quad d_k^T \nabla f(x_{k+1}) \tag{1.26}$$

d'après (1.20), il en résulte

$$f^{'}(\alpha_k) \quad = \quad d_k^T (G x_{k+1} - b) \tag{1.27}$$

remplaçons la formule de x_{k+1} par (1.21)

$$f^{'}(\alpha_k) \quad = \quad d_k^T (G(x_k + \alpha_k d_k) - b) \tag{1.28}$$

comme $f^{'}(\alpha_k) = 0$, nous obtenons alors

$$d_k^T (G(x_k + \alpha_k d_k) - b) \quad = \quad 0, \tag{1.29}$$

enfin,

$$\alpha_k = -\frac{d_k^T g_k}{d_k^T G d_k},$$

avec $g_k = \nabla f(x_k) = G x_k - b$.

Théoreme 5. .*[49]*
Considérons le problème (1.19) avec la matrice G symétrique définie positive. Soit $x_0 \in \mathbb{R}^n$ un vecteur initial, alors la suite générée par l'algorithme du gradient conjugué dans le cas quadratique appliqué au problème (1.19), converge vers la solution optimale x^* en n itérations (i.e. $x_n = x^*$ et $G x_n = G x^* = b$.)

1.4.2 La méthode de gradient conjugué non linéaire

La méthode du Gradient conjugué a été découverte en 1952 par Hestenes et Steifel [31], pour la minimisation des fonctions quadratiques. Plusieurs chercheurs ont étendu cette méthode au cas non linéaire (non quadratique). Cela a été fait pour la première fois en 1964 par Fletcher et Reeves [23].

Le principal avantage des algorithmes de gradient conjugué non linéaires, est leur faible besoin de stockage ainsi leur capacité à résoudre des problèmes d'optimisation non linéaire sans contrainte à grande taille,

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.30}$$

où, $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable.

Tout algorithme non linéaire du gradient conjugué, génère une suite itérative présenté par la relation suivante:

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k, \tag{1.31}$$

le pas α_k est déterminé par une recherche linéaire exacte/ inexacte et d_k est la direction de recherche, calculée de façon récurrente par la relation,

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 0, \tag{1.32}$$

avec $g_k = \nabla f(x_k)$ et $\beta_k \in \mathbb{R}$.

Les différentes valeurs de scalaires β_k définissent les différentes algorithmes du gradient conjugué.

Quelques résultats de convergence des méthodes du gradient conjugué non linéaire

L'objectif est de de déterminer les conditions ou les critères, qui garantissent la convergence de la suite $\{x_k\}$ générée par l'algorithme de gradient conjugué décrit par les relations (1.30) et (1.31), de manière à ce qu'elle converge vers x^*, solution du problème (1.32).

Une condition importante pour les algorithmes d'optimisation basées sur la règle de recherche linéaire, c'est que la direction de la recherche doit être une direction de descente, cela implique que les propriétés suivantes doivent être vérifiées par cette direction, ainsi

$$g_k^T d_k < 0, \quad \text{propriété de descente} \tag{1.33}$$

$$g_k^T d_k < c \parallel g_k \parallel^2, \quad \text{propriété de descente suffisante} \tag{1.34}$$

pour tout $k \geq 0$, où $c > 0$ est une constante.
Rappelons que, la condition de descente suffisante est plus forte que la condition de descente, puisque on a

$$g_k^T d_k < c \parallel g_k \parallel^2 \Longrightarrow g_k^T d_k < 0$$

Pour analyser les propriétés de convergence d'une méthode de recherche linéaire, telle que le gradient conjugué non linéaire, il est possible d'évaluer l'efficacité de la direction de recherche et de la longueur du pas de recherche. L'évaluation de la qualité de la direction de recherche d_k, peut être réalisée en examinant l'angle entre la direction de descente la plus pente $-g_k$ et la direction de recherche d_k. Cet angle est donné par la formule

$$\cos(\theta_k) = \frac{-g_k^T d_k}{\parallel g_k \parallel \parallel d_k \parallel} \tag{1.35}$$

Afin d'établir les résultats généraux de convergence de toute méthode décrit par les relations (1.30) et (1.31), nous introduisons des hypothèses (voir [49]) sur la fonction f à savoir

1. *Hypothèse de bornitude:* L'ensemble $\mathcal{H} = \{x \in R^n / f(x) \leq f(x_0)\}$ est borné, où x_0 est le vecteur initial, (i.e. $\exists A > 0$ pour que $\parallel x \parallel \leq A \ \forall x \in \mathcal{H}$), avec la norme $\parallel . \parallel$ est $\parallel . \parallel_2$ de $\mathbb{R}^n$.

2. *Hypothèse de Lipschitz:* Dans un certain voisinage V de l'ensemble $\mathcal{H}$, f est continument différentiable et son gradient est Lipschitz continu, $\exists B > 0$ telle que

$$\parallel g(x) - g(y) \parallel \leq B \parallel x - y \parallel \qquad \forall x, y \in V. \tag{1.36}$$

Selon les hypothèses de bornitude et de Lipschitz ci-dessus, le théorème suivant, dû à Zoutendijk [60] et Wolfe [51, 52], est essentiel pour démontrer les résultats de convergence globale des algorithmes d'optimisation sans contrainte dans le cas non linéaire, y compris le gradient conjugué ou l'algorithme de Newton.

Théoreme 6. .
Supposons que f est bornée dans $\mathbb{R}^n$ et que f est continûment différentiable dans un voisinage V de l'ensemble $\mathcal{H}$. Supposons aussi que le gradient est Lipschitz continu, i.e. (1.36) est satisfait.
Considérons toute itération de la forme (1.30), où d_k est une direction de descente et α_k vérifie les conditions de recherche linéaire de Wolfe (1.17).
Alors on a

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \parallel g_k \parallel^2 < \infty. \tag{1.37}$$

La relation (1.37) est appelée condition de Zoutendijk, et d'après (1.35), elle peut s'écrire comme suit,

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\parallel d_k \parallel^2} < \infty. \tag{1.38}$$

1.4.3 Quelques méthodes classiques et modifiées de gradient conjugué

La méthode de Fletcher-Reeves

Il s'agit d'une méthode de gradient conjugué connue sous le nom de méthode FR, introduite par Fletcher-Reeves et développée dans la référence [23], où le paramètre β_k dans (1.32) est noté β_k^{FR} et prend la forme du quotient de la norme comme suit

$$\beta_k^{FR} = \frac{\parallel g_{k+1} \parallel^2}{\parallel g_k \parallel^2}$$

En 1970, Zoutendijk [60] a démontré la convergence globale de la méthode FR, lorsqu'elle est associée à la recherche linéaire exacte. En 1977, Powel [47] a remarqué la sensibilité numérique de la méthode FR, lorsqu'elle est associée à une recherche linéaire exacte, puisque l'algorithme peut souvent effectuer de nombreux mesures courtes sans réaliser de progrès significatifs vers le minimum de la fonction. La mauvaise performance observée de la méthode FR dans plusieurs applications, peuvent être largement attribuées à ce phénomène de "brouillage". Al-Baali [41] a étendu le résultat de la méthode FR à la recherche linéaire inexacte forte de Wolfe avec $\delta < 0.5$ dans (1.18). Il a montré que si $\delta < 0.5$, la condition suffisante est vérifiée et il y a une convergence globale.

La méthode de Polak-Ribiére

En 1969, Polak et Ribiére [45] et Polyak [46] ont proposé un autre choix du coefficient du gradient conjugué β_k , noté β_k^{PRP}, qui est donné par la formule,

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\parallel g_k \parallel^2}$$

La convergence globale de la méthode PRP lorsque f est fortement convexe et la recherche linéaire utilisée est exacte, a été démontrée dans [45], . Dans le cas d'une fonction non linéaire générale , Powell [47] a démontré que si la taille du pas $s_k = x_{k+1} - x_k$, tend vers 0 (lorsque la recherche linéaire utilisée est exacte et la continuité Lipschitzienne est assurée), la méthode PRP converge globalement.

En revanche, Powell a développé plus tard [48], un contre-exemple à trois variables, qui démontre que la méthode PRP peut boucler indéfiniment sans approcher aucune solution; par conséquent, la condition selon laquelle la taille du pas tend vers 0, est nécessaire pour garantir la convergence de la méthode. Dans le cas où la direction de recherche est une direction de descente, la convergence globale de l'algorithme PRP, pour les fonctions fortement convexes associées à une recherche linéaire inexacte de Wolfe, a été établie par Yuan [57].

La méthode de Dai-Yuan

C'est une méthode de gradient conjugué qui a été découverte par Y. H. Dai et Y. Yuan [15], connue sous le nom de méthode DY où le paramètre β_k , noté β_k^{DY}, prend la forme suivante

$$\beta_k^{DY} = \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k}$$

La méthode DY, assure toujours la propriété de descente, lorsqu'elle est utilisée avec la recherche linéaire inexacte de Wolfe standard. De plus, si l'hypothèse de Lipschitz est satisfaite, la méthode DY converge globalement vers la solution du problème. Dans [17], Dai a découvert une propriété importante concernant la méthode de DY, qui permet d'établir un lien entre les directions de descente et la condition de descente suffisante. Il a démontré que, si $\exists \gamma_1, \gamma_2$ tel que $\gamma_1 \leq \| g_k \| \leq \gamma_2$, $\forall k \geq 0$ alors $\forall p \in]0, 1[$, $\exists c > 0$ tel que, $g_k^T d_k \leq c \| g_k \|^2$ est vérifiée.

La méthode de Hestenes-Steifel

En 1952, Hestenes et Steifel [31] ont proposé la version linéaire de cette méthode, pour résoudre les systèmes linéaires. Le paramètre β_k , noté β_k^{HS}, s'écrit

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$$

La méthode HS vérifie toujours la condition de conjugaison $d_{k+1}^T y_k = 0$ indépendamment de la recherche linéaire utilisée.
Avec la recherche linéaire exacte on a $\beta_k^{PRP} = \beta_k^{HS}$; par conséquent, la méthode HS est identique à la méthode PRP. Le contre-exemple de Powell [48] pour la méthode PRP, est également applicable à la méthode HS, ce qui montre la non-convergence de la méthode HS avec la recherche de lignes exacte.

La méthode CD

La méthode de descente conjuguée CD proposée par Fletcher [24], où le paramètre β_k , noté β_k^{CD}, prend la forme suivante:

$$\beta_k^{CD} = \frac{\| g_{k+1} \|^2}{-d_k^T g_k}$$

Cette méthode est étroitement liée à la méthode FR. En plus on a eu utilisant une recherche linéaire exacte, $\beta_k^{CD} = \beta_k^{FR}$. Une différence importante entre les algorithmes FR et CD est que, avec CD la condition de descente suffisante, est garantie pour une recherche linéaire de Wolfe forte et la contrainte $\delta \leq \frac{1}{2}$ avec FR, n'est pas nécessaire pour l'algorithme de CD. De plus, pour une recherche linéaire qui vérifie les conditions de Wolfe généralisées, introduites dans [18] avec $\delta_1 < 1$ et $\delta_2 = 0$, il est possible de montrer que $0 \leq \beta_k^{CD} \leq \beta_k^{FR}$. Ansi, d'après l'analyse de [41] ou par le théorème de Zoutendijk [60], la convergence globale est atteinte.

La méthode Dai Liao

Dai et Liao [16] ont introduit une condition de conjugaison généralisée $d_{k+1}^T y_k = -t g_{k+1}^T s_k$ et ont proposé le choix suivant pour β_k à savoir

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k}, \quad t \geq 0.$$

La formule ci-dessus peut être considérée comme une forme modifiée de β_k^{HS}. Dai et Liao [16] ont montré que la méthode CG avec β_k^{DL} est globalement convergente pour pour les fonctions uniformément convexes. Les expériences numériques ont montré que la méthode DL avec t = 0,1 est une amélioration significative de la méthode HS. Pour une recherche de ligne exacte, la méthode DL se réduit aux méthodes HS et PRP, ce qui indique que la méthode DL peut ne pas converger pour les fonctions générales.

La méthode WYL

Wei et al [61] et Huang et al [33], ont donné une formule modifiée pour β_k^{PRP}, pour améliorer les performances et les résultats de convergence de la méthode PRP. Cette formule est noté par β_k^{WYL} et s'écrit

$$\beta_k^{WYL} = \frac{g_{k+1}^T \bar{y_k}}{\| g_k \|^2}$$

. avec $\bar{y_k} = g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k$.
La méthode avec la formule β_k^{WYL} a non seulement de bons résultats numériques, mais possède également la condition de descente suffisante et des propriétés de convergence globale, si la recherche linéaire de Wolfe Powell forte est utilisée avec $\delta \leq \frac{1}{4}$.

Remarque 5. *Si la fonction f est quadratique strictement convexe avec une recherche linéaire exacte, toutes ces variantes de β_k ont la même valeur $\beta_k^{PRP} = \beta_k^{HS} = \beta_k^{FR} = \beta_k^{DY} = \beta_k^{CD}$, de plus la suite engendrée par l'algorithme de gradient conjugué converge en n itération.*

1.4.4 Méthodes hybrides du gradient conjugué

Les méthodes hybrides du gradient conjugué sont développées en combinant différentes méthodes classiques, pour construire de nouvelles méthodes pratiques, qui bénéficient des avantages de chaque méthode combinée.
Les méthodes standard de gradient conjugué peuvent être combinées de deux façons différentes, ce qui conduit à obtenir deux catégories de méthodes hybrides de gradient conjugué.

La première catégorie concerne les méthodes hybrides de gradient conjugué, basées sur le concept de projection, caractérisées par une expression algébrique simple. Lorsqu'une méthode de gradient conjugué entre dans une phase de blocage, le schéma hybride active une autre méthode de gradient conjugué de la paire, dans le but d'améliorer les performances numériques de la méthode hybride. Par exemple, Touati-Ahmed et Storey [50] ont développé l'hybridation suivante:

$$\beta_k = \begin{cases} \beta_k^{PRP} & 0 \leq \beta_k^{PRP} \leq \beta_k^{FR} \\ \beta_k^{FR} & \text{Sinon} \end{cases}$$

lorsque les itérations sont coincées ou bloquées, le paramètre de mise à jour β_k^{PRP} est utilisé. Par les mêmes motivations Hu et Storey [32] ont développé la méthode hybride suivante

$$\beta_k = \max\{0, \min \beta_k^{PRP}, \beta_k^{FR}\}$$

Les expériences numériques confirment que les performances de ces méthodes hybrides, sont souvent meilleures que celles des méthodes classiques.

La deuxième catégorie est celle des méthodes hybrides de gradient conjugué, qui reposent sur la combinaison convexe de méthodes standard. Cela signifie que les méthodes standard de gradient conjugué sont combinées de manière convexe pour former une nouvelle méthode hybride. En général, ces méthodes hybrides sont plus performantes et plus fiables que les méthodes standard; Par exemple, N. Andrei [4] a introduit une méthode de gradient conjugué, nommée CCOMB, qui combine de manière convexe les méthodes de gradient conjugué (PRP) et (DY). Dans le but de bénéficier à la fois des propriétés de calcul de la méthode PRP et des propriétés de convergence de la méthode DY, où le scalaire β_k est donné par la relation suivante:

$$\beta_k^N = (1-\theta_k)\beta_k^{PRP} d_k + \theta_k \beta_k^{DY} \tag{1.39}$$

avec le paramètre de l'hybridation $0 \leq \theta_k \leq 1$, obtenu en utilisant la condition de conjugaison ($d_{k+1}^T y_k = 0$).
Andrei a démontré que les directions générées par cette méthode, sont des directions de descente et que cette méthode est convergente particulièrement sous la condition de descente suffisante.
Un certain nombre de mathématiciens, ont été motivés par l'idée d'Andrei et ont développé de nouvelles méthodes, basées sur la combinaison convexe.

1.5 La méthode de Newton

Considérons le problème de minimisation (1.3),

$$\min_{x \in \mathbb{R}^n} f(x),$$

La méthode de Newton pour l'optimisation sans contrainte, consiste à utiliser itérativement l'approximation quadratique de la fonction objective f à l'itération actuelle x_k et à minimiser cette approximation et prendre sa solution x_{k+1} comme nouvelle approximation du point minimum x^*

Soit f: $\mathbb{R}^n \mapsto \mathbb{R}$ deux fois continument différentiable, $x_k \in \mathbb{R}^n$ et la matrice Hessienne $\nabla^2 f(x)$ défini positif. La modélisation de f au point actuel x_k, se fait par l'approximation quadratique suivante:

$$f(x) \approx f(x_k) + \nabla f(x_k)^T(x-x_k) + \frac{1}{2}(x-x_k)^T \nabla^2 f(x_k)(x-x_k) \tag{1.40}$$

La minimisation de (1.40) conduit à,

$$\nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1}-x_k) = 0 \tag{1.41}$$

alors on obtient la formule itérative suivante:

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k), \tag{1.42}$$

connue sous le nom de méthode de Newton [13], où le vecteur $-\nabla^2 f(x_k)^{-1} \nabla f(x_k)$ est appelée la direction de descente Newton, en utilisant $g_k = \nabla f(x_k)$ cette direction s'écrit alors

$$d_k = -\nabla^2 f(x_k)^{-1} g_k \tag{1.43}$$

Il est clair que si la matrice hessienne est définie positive, alors la direction de Newton est une direction de descente ($g_k d_k = -g_k \nabla^2 f(x_k)^{-1} g_k < 0$)

La méthode de Newton peut converger vers le minimum d'une fonction quadratique en une seule itération; par contre, pour une fonction non quadratique générale, il n'est pas certain que la méthode de Newton puisse atteindre le minimum, avec un nombre fini d'itérations. La méthode de Newton possède une propriété très intéressante qui consiste à, sous certaines hypothèses, un taux de convergence quadratique local est prouvé, ce qui signifie qu'au voisinage de la solution optimale, les erreurs $e_k = \| x_k - x^* \|$ (x^* est la solution optimale), satisfont l'inégalité $e_{k+1} \leq \mu e_k^2$, pour un certain $\mu > 0$ positif.

Le théorème suivant prouve la convergence locale et le taux de convergence quadratique de la méthode de Newton.

Théoreme 7. *.[12]*
Soit $f \in C^2$ et x_k assez proche de la solution x^* du problème de minimisation, avec $\nabla f(x^*) = 0$.
Si la matrice Hessienne $\nabla^2 f(x^*)$ est définie positive et $\nabla^2 f(x^k)$ vérifie la condition de Lipschitz,

$$|\nabla^2 f(x)_{ij} - \nabla^2 f(y)_{ij}| \leq \beta \| x - y \|, \quad \forall \quad 1 \leq i, j \leq n \tag{1.44}$$

où, $\nabla^2 f(x)_{ij}$ est l'élément (i, j) de $\nabla^2 f(x)$, et $\beta > 0$, alors $\forall k \geq 0$, l'itération de Newton (1.42) est bien définie et la séquence générée $\{x_k\}_{k \geq 0}$ converge vers x^* avec un taux de convergence quadratique.

La méthode de Newton avec recherche linéaire

Il est important de noter que la méthode de Newton est une méthode locale, ce qui signifie que si le point de départ est loin de la solution optimal, il n'est pas certain que la matrice $\nabla^2 f(x)$ soit définie positive et que la direction de Newton soit une direction de descente. Par conséquent, la convergence n'est pas garantie. Nous savons que la recherche linéaire est une stratégie de globalisation, nous pouvons donc définir la méthode de Newton avec recherche linéaire (également appelée *"the damped Newton method"*), qui assure la convergence globale. Toutefois il convient de noter que la méthode de Newton ne converge avec un taux quadratique, que lorsque la séquence de pas $\{\alpha_k\}_{k \geq 0}$ converge vers (l'unité)[49].

L'itération de Newton avec recherche linéaire se fait comme suit,

$$d_k = -\nabla^2 f(x_k)^{-1} g_k, \tag{1.45}$$

et

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.46}$$

où, le pas α_k est déterminée en utilisant une technique de recherche linéaire exacte ou inexacte. Le théorème suivant indique la convergence globale de l'algorithme de Newton avec la recherche linéaire exacte.

Théoreme 8. *.[49]*
Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ deux fois continûment différentiable sur un ensemble convexe ouvert $\Omega \subseteq \mathbb{R}^n$. Supposons que $\forall x_0 \in \mathbb{R}^n$, il existe une constante $p > 0$ telle que $f(x)$ satisfasse

$$v^T \nabla^2 f(x) v \geq p \| v \|^2, \quad \forall v \in \mathbb{R}^n, x \in L(x_0) \tag{1.47}$$

où, $L(x_0) = \{x : f(x) \leq f(x_0)\}$ est l'ensemble de niveaux correspondant.
Alors la séquence $\{x_k\}_{k \geq 0}$ générée par l'Algorithme de Newton avec recherche linéaire exacte satisfait les résultats suivantes:

(i) Si la séquence $\{x_k\}$ est finie, alors $\nabla f(x_k) = 0$ pour un certain k.

(ii) Si la séquence $\{x_k\}$ est infinie, alors $\{x_k\}$ converge vers l'unique minimum x^* de f.

Maintenant, étudions la convergence de la méthode de Newton avec la recherche linéaire inexacte. Si la recherche linéaire de Wolfe faible est appliquée, alors il s'en suit que

$$f(x_k) - f(x_x + \alpha_k d_k) \geq \eta_k \parallel d_k \parallel^2 cos^2 < d_k, -g_k >, \tag{1.48}$$

où, η_k est une constante indépendante de k.
Alors, la convergence globale de la méthode de Newton avec les condition de la recherche linéaire inexacte, est prouvée dans le théorème suivant:

Théoreme 9. *.[49]*
Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ deux fois continûment différentiable sur un ensemble convexe ouvert $\Omega \subseteq \mathbb{R}^n$. Supposons que $\forall x_0 \in \mathbb{R}^n$, il existe une constante $p > 0$ telle que $f(x)$ satisfasse la condition (1.47) sur l'ensemble de niveau $L(x_0)$.
Si la recherche linéaire satisfait (1.48), alors la séquence $\{x_k\}_{k\geq 0}$ générée par l'Algorithme de Newton avec une recherche linéaire inexacte satisfait

$$\lim_{k\mapsto\infty} \parallel g_k \parallel = 0, \tag{1.49}$$

de plus $\{x_k\}$ converge vers l'unique minimum x^* de la fonction f.

Remarque 6. *.[49]*

Nous constatons que

1. *l'avantage principale de la méthode de Newton est qu'elle bénéficie d'une convergence quadratique*
2. *La méthode de Newton est très bien adaptée aux problèmes de faible dimension, pour lesquels il est facile de calculer la matrice hessienne et son inverse. Par contre, pour les problèmes de grande taille, le calcul de la matrice hessienne et son inverse peut s'avérer complexe et coûteux, de plus les problèmes de convergence peuvent être plus fréquents.*
3. *Présente une certains sensibilité aux conditions initiales, puisque lorsque le choix initial est éloigné de la solution, la méthode de Newton peut soit diverger, soit converger vers un autre point (pas un minimum: maximum ou point selle).*

Ceci conduit à conclure que la méthode de Newton, ne génère pas généralement une suite qui converge vers le minimum. Cependant, sous certaines conditions (Hessien définie positif, le point initial suffisamment proche de la solution optimale,...), elle devient très intéressante et converge rapidement avec un taux de convergence quadratique.

La méthode Quasi-Newton est une amélioration de la méthode de Newton qui vise à surmonter certains de ses inconvénients. Cette méthode s'inspire de la méthode de Newton, mais sans avoir besoin de calculer la matrice hessienne ou son inverse, l'idée est donc de remplacer l' itération de la méthode de Newton par

$$x_{k+1} = x_k - \alpha_k B_k g_k, \qquad k \geq 0, \tag{1.50}$$

où, α_k est le pas de recherche choisis à l'aide des conditions de recherche linéaire, le long de la direction $d_k = -B_K g_k$, B_k est une approximation symétrique définie positif de l'inverse de la matrice Hessienne.
L'objectif principale est donc de trouver une bonne suite de matrices B_k, facile à construire qui converge rapidement vers des approximations précises de l'inverse du Hessien.
Prenons $f \in C(\mathbb{R}^n)$ et faisons un développement de $\nabla f(x)$ au voisinage de x_k,

$$\begin{aligned} \nabla f(x) &= \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) + o(\parallel x - x_k \parallel) \\ &\approx \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) \end{aligned}$$

ou encore

$$\nabla^2 f(x_k)^{-1}[\nabla f(x) - \nabla f(x_k)] \approx x - x_k$$

Les approximations sons exacts si f est une la fonction quadratique. En particulier avec $x = x_{k+1}$ et si B_k est une bonne approximation de l'inverse du Hessien $\nabla^2 f(x_k)^{-1}$, alors

$$B_k[\nabla f(x_{k+1}) - \nabla f(x_k)] \approx x_{k+1} - x_k$$

Nous pouvons imposer que B_{k+1} satisfasse exactement cette équation, donc

$$B_{k+1}[\nabla f(x_{k+1}) - \nabla f(x_k)] = x_{k+1} - x_k \tag{1.51}$$

cette équation s'appelle l'équation de la sécante [59].
A l'étape k, la mise à jour de la matrice d'approximation se fait de manière simple à l'aide de la formule suivante:

$$B_{k+1} = B_k + D_k,$$

où, D_k est une matrice de correction, calculée de telle manière qu'elle intègre de manière optimale les nouvelles informations fournies par x_{k+1} et $\nabla(x_{k+1})$, de telle sorte que B_{k+1} satisfasse la condition donnée par (1.51). En se basant sur ce concept, on peut conclure que les méthodes de Quasi-Newton se distinguent les unes des autres, en fonction de la définition de la matrice D_k.

Chapitre 2

Un nouvel algorithme hybride de gradient conjugué basé sur la direction de Newton pour l'optimisation sans contrainte

Introduction

Dans cette étude, nous considérons les problèmes d'optimisation sans contrainte, formulés comme suit

$$\min_{x\in\mathbb{R}^n} f(x), \tag{2.1}$$

où, f: $\mathbb{R}^n \mapsto \mathbb{R}$ est une fonction continûment différentiable et son gradient est noté $g(x) = \nabla f(x)$. Les techniques numériques pour résoudre (4.1) sont itératives, c'est-à-dire qu'à partir d'un vecteur initial approprié $x_0 \in \mathbb{R}^n$, les itérations sont générées par la relation de récurrence suivante

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k \geq 0, \tag{2.2}$$

où, α_k est la taille du pas déterminée à l'aide d'une technique de recherche linéaire exacte/inexacte et d_k est la direction de recherche, supposée satisfaire la propriété de descente

$$g_k^T d_k < 0, \quad k \geq 0,$$

ou la condition de descente suffisante

$$g_k^T d_k \leq -C \parallel g_k \parallel^2, \quad k \geq 0, \quad C > 0.$$

La taille du pas α_k, est généralement choisie pour satisfaire la célèbre recherche linéaire inexacte de Wolfe forte suivante [51, 52]

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta\alpha_k g_k^T d_k \tag{2.3}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{2.4}$$

où, $0 < \delta < \sigma < \frac{1}{2}$.

En fonction du calcul de la direction de recherche, il existe plusieurs méthodes pour résoudre (4.1). Dans cette étude nous nous intéressons à la méthode de Newton et aux méthodes du gradient conjugué, la direction de recherche de la méthode de Newton est calculée comme suit

$$d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}, \tag{2.5}$$

avec $\nabla^2 f(x_{k+1})$ est la matrice hessienne de f.
La méthode de Newton utilise l'information de la dérivée seconde (matrice hessienne), pour mettre à jour la direction de recherche d_k, ce qui lui permet d'obtenir un taux de convergence quadratique, mais en pratique, surtout lorsque n est grand, les méthodes qui ne nécessitent pas d'évaluation de la hessienne sont préférées à celles qui en ont besoin [11].
La méthode du gradient conjugué ne nécessite pas beaucoup d'espace de stockage par rapport à d'autres méthodes, car elle n'a besoin que de l'information sur la dérivée première, ce qui la rend très pratique pour résoudre des problèmes d'optimisation sans contrainte à grande échelle [43], où

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k d_k. \tag{2.6}$$

Selon le choix du paramètre $\beta_k \in \mathbb{R}$, connu sous le nom de paramètre du gradient conjugué, il existe plusieurs algorithmes de gradient conjugué différents. Dans ce qui suit, nous rappelons quelques formules célèbres pour ce paramètre, ainsi

$$\begin{aligned}
\beta_k^{HS} &= \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \text{(HS - Hestenes et Stiefel [31])},\\
\beta_k^{FR} &= \frac{\| g_{k+1} \|^2}{\| g_k \|^2}, \quad \text{(FR - Fletcher et Reeves [23])},\\
\beta_k^{PRP} &= \frac{g_{k+1}^T y_k}{\| g_k \|^2}, \quad \text{(PRP - Polak et Ribiére [45, 46])},\\
\beta_k^{CD} &= \frac{\| g_{k+1} \|^2}{-d_k^T g_k}, \quad \text{(CD - conjugate descent [24])},\\
\beta_k^{LS} &= \frac{g_{k+1}^T y_k}{-d_k^T g_k}, \quad \text{(LS - Liu et Storey [40])},\\
\beta_k^{DY} &= \frac{\| g_{k+1} \|^2}{d_k^T y_k}, \quad \text{(DY - Dai et Yuan[15])},\\
\beta_k^{DL} &= \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k}, \quad \text{(DL - Dai et Liao [16])}.
\end{aligned}$$

où, $\| \,.\, \|$ est la norme euclidienne, $t \geq 0$, $y_k = g_{k+1} - g_k$ et $s_k = \alpha_k d_k$.

De nombreux chercheurs ont essayé de concevoir de nouvelles méthodes, basées sur des techniques hybrides, qui sont considérées comme plus efficaces que les méthodes originales, car elles visent à intégrer les points forts et les bonnes performances des méthodes à combiner. Plusieurs méthodes hybrides ont donc été proposées, par exemple, Xu et Kong [54] ont proposé deux méthodes hybrides, la première étant une combinaison linéaire entre les méthodes de gradient conjugué DY et HS et la seconde entre FR et PRP. Djordjevic [19] a proposé une méthode hybride de gradient conjugué, en utilisant la combinaison convexe des méthodes FR et PRP, où la direction de recherche satisfait la condition de conjugaison. Andrai [6] a présenté une autre méthode hybride de gradient conjugué avec β_k calculé comme une combinaison convexe des méthodes DY et HS, La particularité de cette méthode hybride, est que la direction de recherche représente la direction de Newton et qu'elle surpasse de nombreuses autres méthodes de gradient conjugué. Cette idée a inspiré de nombreux chercheurs pour concevoir de nouvelles méthodes hybrides, voir [37, 39, 21].

Dans ce travail, nous visons à tirer parti de certaines propriétés souhaitables des algorithmes du gradient conjugué et de Newton tout en évitant leurs inconvénients, en intégrant la direction de Newton avec la direction du gradient conjugué. Pour ce faire, motivés par l'idée d'Andrai[6], nous proposons de combiner les direction de gradient conjugué DY et DL, sous la forme d'une combinaison convexe, pour créer une nouvelle direction hybride basée sur la direction de Newton.

l'algorithme vise à atteindre numériquement la convergence quadratique de la méthode de Newton, sans avoir besoin d'évaluer ou stoker la matrice Hessienne, tout en conservant certaines bonnes propriétés de l'algorithme de gradient conjugué, telles que ses avantages computationnels, sa capacité de résoudre des problèmes à grand échelles et sa convergence globale.

2.1 La nouvelle méthode hybride du gradient conjugué

Dans cette section, nous présentons notre méthode hybride de gradient conjugué, comme une combinaison convexe des algorithmes DY et DL en définissant la direction de recherche comme suit

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k^{DYDL} d_k, \tag{2.7}$$

où

$$\beta_k^{DYDL} = (1-\delta_k)\beta_k^{DL} + \delta_k \beta_k^{DY}. \tag{2.8}$$

On peut donc écrire

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + (1-\delta_k)\beta_k^{DL} d_k + \delta_k \beta_k^{DY} d_k, \tag{2.9}$$

où $\delta_k \in [0,1]$. Si $\delta_k = 0$, alors $\beta_k^{DYDL} = \beta_k^{DL}$ et si $\delta_k = 1$, alors $\beta_k^{DYDL} = \beta_k^{DY}$. En revanche, si $0 < \delta_k < 1$ alors β_k^{DYDL} est la combinaison convexe entre β_k^{DY} et β_k^{DL}.

Supposons que $\nabla^2 f(x)^{-1}$ existe à chaque point itératif pour la fonction objective f.
L'objectif principal de cet algorithme, est d'approcher numériquement le comportement de convergence quadratique, observée dans la méthode de Newton, tout en conservant certaines caractéristiques essentielles de l'algorithme de gradient conjugué. Pour ce faire, nous incorporons la direction de Newton avec la direction du gradient conjugué, par calculer le scalaire δ_k de sorte que notre direction de recherche donnée par (2.9) soit égale à la direction de Newton.

$$-g_{k+1} + (1-\delta_k)\beta_k^{DL} d_k + \delta_k \beta_k^{DY} d_k = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}. \tag{2.10}$$

Cette idée est similaire à celle d'Andrai, voir [6].

En multipliant les deux côtés de l'équation (2.10) par $s_k^T \nabla^2 f(x_{k+1})$ à partir de la gauche, nous obtenons,

$$-s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (1-\delta_k)\beta_k^{DL} s_k^T \nabla^2 f(x_{k+1}) d_k + \delta_k \beta_k^{DY} s_k^T \nabla^2 f(x_{k+1}) d_k = -s_k^T g_{k+1}. \tag{2.11}$$

Supposons que la paire (s_k, y_k) satisfasse à la condition de sécante,

$$\nabla^2 f(x_{k+1}) s_k = y_k,$$

qui s'écrit

$$s_k^T \nabla^2 f(x_{k+1}) = y_k^T.$$

Alors (2.11) devient

$$-y_k^T g_{k+1} + (1-\delta_k)\beta_k^{DL} y_k^T d_k + \delta_k \beta_k^{DY} y_k^T d_k = -s_k^T g_{k+1}.$$

Après avoir effectué quelques calculs algébriques, nous obtenons

$$\delta_k = \frac{-s_k^T g_{k+1} + y_k^T g_{k+1} - \beta_k^{DL} y_k^T d_k}{(-\beta_k^{DL} + \beta_k^{DY}) y_k^T d_k} \tag{2.12}$$

En remplaçant les formules des β_k^{DY} et β_k^{DL} dans (2.12), nous obtenons

$$\delta_k = \frac{-s_k^T g_{k+1} + y_k^T g_{k+1} - g_{k+1}^T y_k + t g_{k+1}^T s_k}{(-g_{k+1}^T (g_{k+1} - g_k) + t g_{k+1}^T s_k + \| g_{k+1} \|^2} \tag{2.13}$$

En poursuivant les simplifications, nous parvenons à l'expression suivante

$$\delta_k = \frac{-s_k^T g_{k+1} + t g_{k+1}^T s_k}{-g_{k+1}^T g_{k+1} + g_{k+1}^T g_k + t g_{k+1}^T s_k + \| g_{k+1} \|^2} \tag{2.14}$$

ce qui donne

$$\delta_k = \frac{s_k^T g_{k+1}(t-1)}{g_{k+1}^T (g_k + t s_k)}. \tag{2.15}$$

Il est clair que, bien que nous ayons calculé le scalaire δ_k de sorte que la direction (2.9) soit la direction de Newton, notre algorithme ne nécessite pas de calculer ou de stocker la dérivée seconde (la matrice hessienne), requise par la méthode classique de Newton, et ce en raison de l'utilisation de l'équation de la sécante.

Nous présentons maintenant notre algorithme DYDL, qui présente des caractéristiques intéressantes de l'algorithme du gradient conjugué et de l'algorithme de Newton.

Algorithme DYDL

Etape 0: Choisir le point initial $x_0 \in \mathbb{R}^n$, $\epsilon > 0$, $t > 1$.
Calculer $f_0 = f(x_0)$ et $g_0 = \nabla f(x_0)$.
Définir $d_0 = -g_0$, l'estimation initiale $\alpha_0 = \frac{1}{\|g_0\|}$. Soit $k = 0$.
Etape 1: Tester un critère pour arrêter les itérations, c'est-à-dire si $\| g_k \| \leq \epsilon$, alors arrêter. Sinon, passer à l'étape 2.
Etape 2: Calculer la taille de pas α_k en utilisant les conditions de strong Wolfe (2.3), (2.4).
Etape 3: Mettre à jour la prochaine itération par $x_{k+1} = x_k + \alpha_k d_k$. Calculer $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$ et $s_k = x_{k+1} - x_k$.
Etape 4: Si $g_{k+1}^T(g_k + ts_k) = 0$, alors $\delta_k = 0$, sinon calculer δ_k comme dans (2.15).
Etape 5: Si $\delta_k \leq 0$, alors calculer $\beta_k^{DYDL} = \beta_k^{DL}$.
Si $\delta_k \geq 1$, alors calculer $\beta_k^{DYDL} = \beta_k^{DY}$.
Si $0 < \delta_k < 1$, alors calculer β_k^{DYDL} comme dans (2.8).
Etape 6: Calculer $d_{k+1} = -g_{k+1} + \beta_k^{DYDL} d_k$. Définir la valeur initiale $\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$.
Etape 7: Soit $k = k + 1$, Revenir à l'étape 1.

Remarque 7. *Si $\beta_k^{DYDL} = \beta_k^{DL}$ ou $\beta_k^{DYDL} = \beta_k^{DY}$, alors dans ces deux cas, on se se réfère à [16] et [15].*

Les deux théorèmes suivants confirment que les conditions de descente et de descente suffisante sont satisfaites.

Théoreme 10. .
Soit δ_k donné par (2.15) avec $0 < \delta_k < 1$. Supposons que $t > 1$ dans (2.15) et que α_k dans l'algorithme DYDL est déterminé par la règle de Wolfe forte (2.3), (2.4).
Alors la direction définie par (2.9) est une direction de descente, (i.e. $g_k^T d_k < 0$.).

Démonstration 1. .
La preuve se fait par récurrence.
Pour k=0 : $g_0^T d_0 = -g_0^T g_0 = -\| g_0 \|^2 < 0$.
Supposons que $g_k^T d_k < 0$ est satisfait pour $k \geq 1$ et montrons qu'il est satisfait pour $k + 1$.
En multipliant (2.9) par g_{k+1}, on trouve

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + (1 - \delta_k)\beta_k^{DL} g_{k+1}^T d_k + \delta_k \beta_k^{DY} g_{k+1}^T d_k, \tag{2.16}$$

En remplaçant les formules des β_k^{DY} et β_k^{DL} dans (2.16), nous obtenons

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + (1 - \delta_k)\frac{g_{k+1}^T(y_k - ts_k)}{d_k^T y_k} g_{k+1}^T d_k + \delta_k \frac{\| g_{k+1} \|^2}{d_k^T y_k} g_{k+1}^T d_k, \tag{2.17}$$

sachant que $y_k = g_{k+1} - g_k$, la relation (2.17) devient

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + (1 - \delta_k)\frac{g_{k+1}^T((g_{k+1} - g_k) - ts_k)}{d_k^T y_k} g_{k+1}^T d_k + \delta_k \frac{\| g_{k+1} \|^2}{d_k^T y_k} g_{k+1}^T d_k,$$

En simplifiant les calculs, nous arrivons à

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\| g_{k+1} \|^2 + \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k \\ &- \delta_k \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \delta_k \frac{\| g_{k+1} \|^2}{d_k^T y_k} g_{k+1}^T d_k, \end{aligned}$$

En poursuivant les simplifications, nous obtenons

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\| g_{k+1} \|^2 + \frac{\| g_{k+1} \|^2 - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k - \delta_k \frac{\| g_{k+1} \|^2}{d_k^T y_k} g_{k+1}^T d_k \\ &- \delta_k \frac{-g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \delta_k \frac{\| g_{k+1} \|^2}{d_k^T y_k} g_{k+1}^T d_k, \end{aligned}$$

ce qui donne

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\| g_{k+1} \|^2 + \frac{\| g_{k+1} \|^2 - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k \\ &+ \delta_k \frac{g_{k+1}^T g_k + t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k. \end{aligned} \tag{2.18}$$

Nous remplaçons δ_k dans la relation (2.18) par (2.15), alors nous obtenons

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{g_{k+1}^T (g_k + t s_k)}{d_k^T y_k} g_{k+1}^T d_k + (t-1)\frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k. \quad (2.19)$$

D'après (2.15) nous avons

$$\frac{g_{k+1}^T (g_k + t s_k)}{s_k^T g_{k+1}(t-1)} = \frac{1}{\delta_k},$$

alors

$$g_{k+1}^T (g_k + t s_k) = \frac{(t-1) s_k^T g_{k+1}}{\delta_k}.$$

Par conséquent, (2.19) devient

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{\delta_k d_k^T y_k} g_{k+1}^T d_k + (t-1)\frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k. \quad (2.20)$$

Puisque $s_k = \alpha_k d_k$, alors

$$\frac{(t-1)\alpha_k (d_k^T g_{k+1})^2}{d_k^T y_k} = \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k. \quad (2.21)$$

En utilisant la recherche linéaire de Wolfe forte (2.4), nous obtenons

$$d_k^T y_k = g(x_k + \alpha_k d_k)^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k > 0. \quad (2.22)$$

Comme $t > 1$ et puisque $\alpha_k > 0$, $d_k^T y_k > 0$ et d'après (2.21) nous concluons que

$$\frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k > 0. \quad (2.23)$$

Sachant que $\frac{1}{\delta_k} > 1$, et d'après (2.23) ceci donne

$$\frac{(t-1) s_k^T g_{k+1}}{\delta_k d_k^T y_k} g_{k+1}^T d_k > \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k, \quad (2.24)$$

en multipliant la relation (2.24) par (-), nous obtenons

$$-\frac{(t-1) s_k^T g_{k+1}}{\delta_k d_k^T y_k} g_{k+1}^T d_k < -\frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k, \quad (2.25)$$

en ajoutant quelques termes aux deux cotés de l'inégalité (2.25), nous obtenons

$$\begin{aligned} & - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{\delta_k d_k^T y_k} g_{k+1}^T d_k + (t-1)\frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k \\ & < -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k + (t-1)\frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k. \end{aligned}$$

Par conséquent, (2.20) devient

$$g_{k+1}^T d_{k+1} \quad < \quad -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k + (t-1)\frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k,$$

après calcules

$$g_{k+1}^T d_{k+1} \quad < \quad -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \quad (2.26)$$

ce qui donne

$$g_{k+1}^T d_{k+1} \quad < \quad -(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k}) \parallel g_{k+1} \parallel^2 . \quad (2.27)$$

puisque $d_k^T y_k > 0$ et $d_k^T g_k < 0$, on a alors

$$g_{k+1}^T d_k = d_k^T y_k + d_k^T g_k < d_k^T y_k,$$

ainsi

$$g_{k+1}^T d_k < d_k^T y_k \Longrightarrow \frac{g_{k+1}^T d_k}{d_k^T y_k} < 1,$$

donc

$$-(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k}) < 0.$$

Par conséquent (2.27), *devient*

$$g_{k+1}^T d_{k+1} < -(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k}) \parallel g_{k+1} \parallel^2 < 0. \tag{2.28}$$

Ce qui confirme que, $g_k^T d_k < 0$.

Théoreme 11. .
Soit δ_k *donné par* (2.15) *avec* $0 < \delta_k < 1$. *Supposons que* $t > 1$ *dans* (2.15) *et que* α_k *dans l'algorithme DYDL est déterminé par la règle de Wolfe forte* (2.3), (2.4).
Alors il existe une constante $C > 0$ *telle que la condition de descente suffisante*

$$g_{k+1}^T d_{k+1} \leq -C \parallel g_{k+1} \parallel^2 \tag{2.29}$$

se vérifie.

Démonstration 2. .
En multipliant (2.9) *par* g_{k+1}, *on trouve*

$$g_{k+1}^T d_{k+1} = - \parallel g_{k+1} \parallel^2 + (1-\delta_k)\beta_k^{DL} g_{k+1}^T d_k + \delta_k \beta_k^{DY} g_{k+1}^T d_k. \tag{2.30}$$

En remplaçant les formules des β_k^{DY} *et* β_k^{DL} *dans (2.30), nous obtenons*

$$g_{k+1}^T d_{k+1} = - \parallel g_{k+1} \parallel^2 + (1-\delta_k)\frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k} g_{k+1}^T d_k + \delta_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \tag{2.31}$$

On reprend le même calcules que nous avons vu dans la démonstration du théorème 10, on obtient

$$g_{k+1}^T d_{k+1} < - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \tag{2.32}$$

ainsi

$$g_{k+1}^T d_{k+1} \leq - \parallel g_{k+1} \parallel^2 + \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \parallel g_{k+1} \parallel^2 . \tag{2.33}$$

D'après la deuxième condition de Wolfe forte (2.4), *il en résulte*

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k \tag{2.34}$$

c'est à dire

$$\sigma g_k^T d_k \leq g_{k+1}^T d_k \leq -\sigma g_k^T d_k.$$

Alors

$$d_k^T y_k = g_{k+1}^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k > 0$$

ce qui donne

$$\frac{1}{d_k^T y_k} \leq \frac{1}{-(1-\sigma) g_k^T d_k}. \tag{2.35}$$

De (2.34) *et* (2.35), *il résulte*

$$\frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \leq \frac{\sigma}{1-\sigma}.$$

Ainsi (2.33) devient

$$g_{k+1}^T d_{k+1} \leq - \parallel g_{k+1} \parallel^2 + \frac{\sigma}{1-\sigma} \parallel g_{k+1} \parallel^2$$

alors

$$g_{k+1}^T d_{k+1} \leq -(1 - \frac{\sigma}{1-\sigma}) \parallel g_{k+1} \parallel^2 .$$

Il suffit de prendre $C = (1 - \frac{\sigma}{1-\sigma}) > 0$, *car* $0 < \sigma < \frac{1}{2}$.
Par conséquent, la condition de descente suffisante est vérifiée.

2.2 Les propriétés de convergence de notre méthode

Les hypothèses essentielles (voir [7]) suivantes sur la fonction objective sont nécessaires pour établir la convergence globale de notre méthode hybride (DYDL).

H1. L'ensemble de niveaux $\mathcal{H} = \{x \in R^n / f(x) \leq f(x_0)\}$ est borné, où x_0 est le vecteur initial.

H2. Dans un certain voisinage $\mathcal{Q}$ de $\mathcal{H}$, la fonction objective f est continument différentiable, son gradient est continu Lipschitz, par conséquent $\exists L > 0$ tel que

$$\| g(x) - g(y) \| \leq L \| x - y \| \qquad \forall x, y \in \mathcal{Q}. \tag{2.36}$$

Ces hypothèses impliquent que, $\exists \gamma \geq 0$ tel que

$$\| g(x) \| \leq \gamma. \quad \forall x \in \mathcal{H} \tag{2.37}$$

Le lemme 1 de Zoutendijk [60] et le Lemme 2 vont utilisé pour démontrer le théorème de convergence ci après.

Lemme 1. .
Supposons que les hypothèses H1 et H2 ci-dessus, soient vérifiés.
Considérons toute itération de la forme (2.2), où d_k satisfait à la condition de descente $g_k^T d_k < 0$ et α_k satisfait à la recherche linéaire inexacte de Wolfe ou à sa version forte.
Alors on a

$$\sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\| d_k \|^2} < +\infty \tag{2.38}$$

Lemme 2. *[39].*
Supposons que les hypothèses H1 et H2 ci-dessus se vérifient. Si d_k est une direction de descente et que la longueur de pas α_k satisfait

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \quad \sigma < 1, \tag{2.39}$$

alors

$$\alpha_k \geq \frac{1-\sigma}{L} \frac{|d_k^T g_k|}{\| d_k \|^2}. \tag{2.40}$$

Ceci montre que α_k obtenu dans la méthode (DYDL) n'est pas égal à zéro, c'est-à-dire qu'il existe une constante $\lambda > 0$ telle que

$$\alpha_k \geq \lambda, \quad \forall k \geq 0. \tag{2.41}$$

Le théorème suivant garantit la convergence globale.

Théoreme 12. .
Supposons que les hypothèses H1, H2 soient vérifiés.
Soit les séquences $\{x_k\}$ générées par notre algorithme (DYDL).
Alors

$$\lim_{k \mapsto \infty} inf \| g_k \| = 0. \tag{2.42}$$

Démonstration 3. .
On va démontrer ce théorème en utilisant un raisonnement par l'absurde.
Supposons que la relation (2.42) n'est pas vérifié. Il existe alors une constante $r > 0$ telle que

$$\| g_k \| > r. \tag{2.43}$$

D'après (2.8), nous avons

$$|\beta_k^{DYDL}| \leq (1-\delta_k)|\beta_k^{DL}| + \delta_k |\beta_k^{DY}|, \tag{2.44}$$

comme $0 < \delta_k < 1$, alors

$$|\beta_k^{DYDL}| \leq |\beta_k^{DL}| + |\beta_k^{DY}|, \tag{2.45}$$

En remplaçant les formules des β_k^{DY} et β_k^{DL}, nous obtenons

$$|\beta_k^{DYDL}| \leq \frac{|g_{k+1}^T (y_k - t s_k)|}{|d_k^T y_k|} + \frac{\| g_{k+1} \|^2}{|d_k^T y_k|}, \tag{2.46}$$

En simplifiant les calculs, nous arrivons à

$$|\beta_k^{DYDL}| \quad \leq \quad \frac{|g_{k+1}^T y_k| + t|g_{k+1}^T s_k)|}{|d_k^T y_k|} + \frac{\| g_{k+1} \|^2}{|d_k^T y_k|}, \tag{2.47}$$

Sachant que $|g_{k+1}^T y_k| <\| g_{k+1} \|\| y_k \|$ *et* $|g_{k+1}^T y_k| <\| g_{k+1} \|\| s_k \|$, *nous obtenons alors*

$$|\beta_k^{DYDL}| \quad \leq \quad \frac{\| g_{k+1} \|\| y_k \| +t \| g_{k+1} \|\| s_k \|}{|d_k^T y_k|} + \frac{\| g_{k+1} \|^2}{|d_k^T y_k|}. \tag{2.48}$$

D'après la deuxième condition de Wolfe forte (2.4) et la condition de descente suffisante (2.29), nous avons

$$d_k^T y_k \geq -(1-\sigma) g_k^T d_k \geq (1-\sigma) C \| g_k \|^2 .$$

D'après (2.43), nous obtenons,

$$d_k^T y_k \geq (1-\sigma) C r^2,$$

donc

$$\frac{1}{d_k^T y_k} \leq \frac{1}{(1-\sigma) C r^2}. \tag{2.49}$$

Sachant que $y_k = g_{k+1} - g_k$ *et* $s_k = x_{k+1} - x_k$, *nous obtenons d'après (2.36),*

$$\| y_k \| \leq l \| s_k \| \leq lD, \tag{2.50}$$

où, $D = \max\{\| x-y \| /x, y \in \mathcal{H}\}$ *est le diamètre de l'ensemble de niveaux* $\mathcal{H}$.
D'après (2.37), (2.49) et (2.50) la relation (2.48) devient,

$$|\beta_k^{DYDL}| \leq \frac{\gamma L D + t\gamma D}{(1-\sigma) C r^2} + \frac{\gamma^2}{(1-\sigma) C r^2} = \frac{\gamma L D + t\gamma D + \gamma^2}{(1-\sigma) C r^2} = E,$$

avec E *est un constant.*
Par conséquent la direction (dans (2.9)) devient,

$$\| d_{k+1} \| \leq \| g_{k+1} \| + |\beta_k^{DYDL}| \| d_k \| \leq \gamma + E \| d_k \| . \tag{2.51}$$

Sachant que $s_k = \alpha_k d_k$ *donc* $\| d_k \| = \frac{\|s_k\|}{\alpha_k}$ *et d'après (2.41), nous obtenons*

$$\| d_{k+1} \| \leq \gamma + E \frac{\| s_k \|}{\alpha_k} \leq \gamma + E \frac{D}{\lambda} = M.$$

avec M *est un constant.*
Ce qui donne alors

$$\frac{1}{\| d_{k+1} \|} \geq \frac{1}{M}.$$

Donc

$$\sum_{k\geq 0} \frac{1}{\| d_{k+1} \|} \geq \sum_{k\geq 0} \frac{1}{M}.$$

Par conséquent,

$$\sum_{k\geq 0} \frac{1}{\| d_k \|^2} = +\infty. \tag{2.52}$$

De plus, à partir de (2.38), (2.43) et (2.29), il en résulte

$$C^2 r^4 \sum_{k\geq 0} \frac{1}{\| d_k \|^2} \leq \sum_{k\geq 0} \frac{C^2 \| g_k \|^4}{\| d_k \|^2} \leq \sum_{k\geq 0} \frac{(g_k^T d_k)^2}{\| d_k \|^2} < +\infty, \tag{2.53}$$

alors

$$\sum_{k\geq 0} \frac{1}{\| d_k \|^2} < +\infty.$$

*Ceci est en contradiction avec (2.52), donc nous avons prouvé (**??**).*

2.3 Expérience numérique

Dans cette section, nous allons discuter des expériences numériques de notre algorithme DYDL en les comparant à celles des algorithmes DY [15] et DL [16]. Pour cela, nous avons 80 problèmes d'optimisation sans contrainte tirés de [8], chaque problème étant testé pour les variables suivantes : 2, 50, 100, 200, 500, 1000, 2000, 3000, 5000 et 10000. Tous les codes sont compilés sur un PC ayant les spécifications suivantes Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50 GHz, 4,00 Go RAM.

Nous présentons les comparaisons numériques avec les autres algorithmes, y compris les profils de performance, donnés par Dolan et Moré [22], dans les conditions de recherche de la ligne de Wolfe forte (2.3), (2.4) avec $\delta = 0.0001$ et $\sigma = 0.1$, et nous utilisons le critère d'arrêt $\| g_k \|_\infty \leq 10^{-7}$, pour tous les algorithmes.

Tout d'abord, nous choisissons la meilleure valeur du paramètre t, Comme le montrent les figures 2.1 et 2.2. Notre algorithme DYDL avec $t = 300$ est plus performant que $t = 2$ et $t = 100$ en termes de temps CPU et de nombre d'itérations.

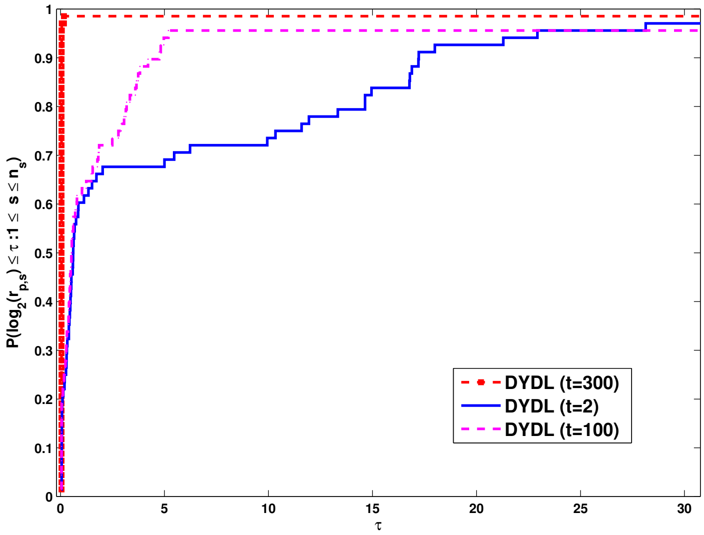


Figure 2.1: Profil de performance basé sur le temps CPU pour choisir la meilleur t de l'algorithme DYDL

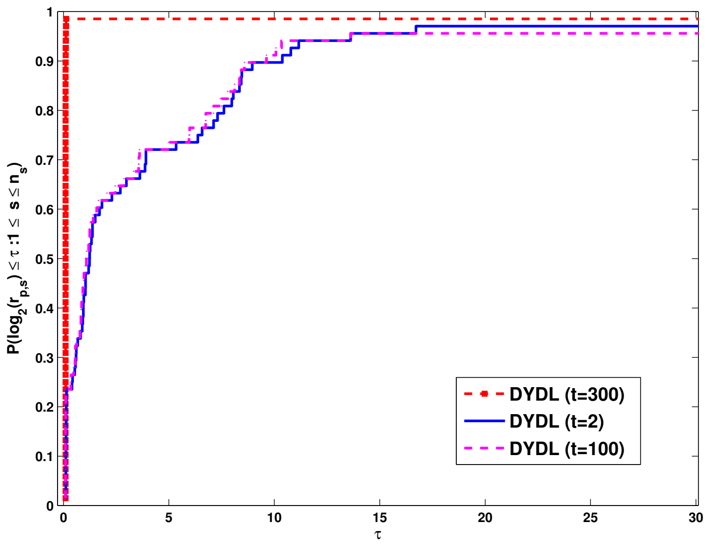


Figure 2.2: Profil de performance basé sur le nombre d' itération pour choisir la meilleur t de l'algorithme DYDL

Notons que, concernant la méthode (DL) proposée par Dai et Liao [16], les résultats numériques ont été appliqués pour $t = 1$, nous comparons donc l'algorithme DYDL pour $t = 300$ avec l'algorithme (DL) pour $t = 1$ et $t = 300$.

Les figures 2.3, 2.4 et 2.5 montrent le profil de performance basé sur le temps CPU, le nombre d'itérations et l'évaluations du gradient, respectivement. Toutes les figures indiquent que les performances de l'algorithme DYDL pour t=300 sont nettement meilleures que celles des algorithmes DY, DL(t=1) et DL(t=300).

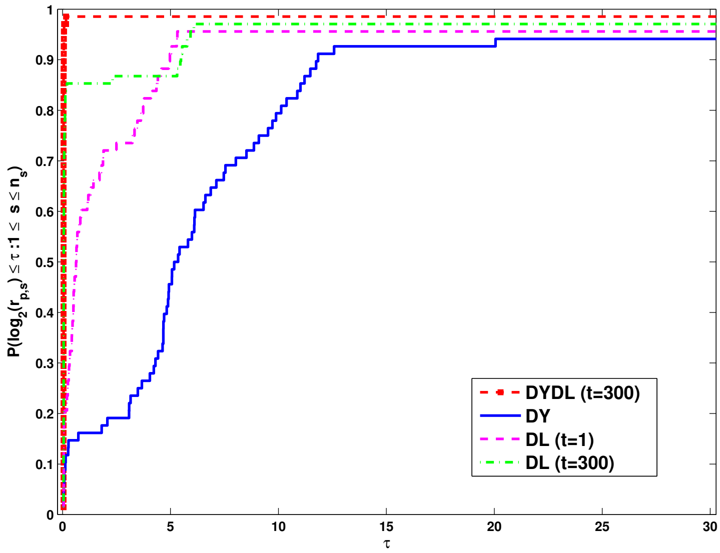


Figure 2.3: Profil de performance basé sur le temps CPU.

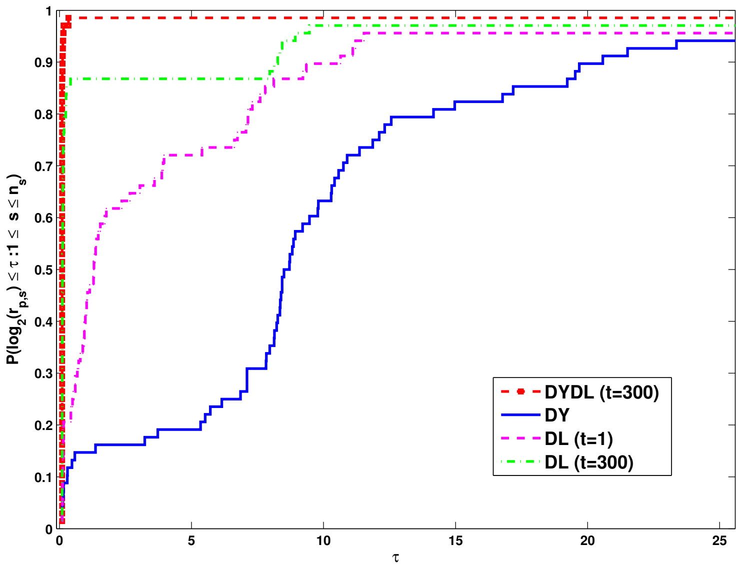


Figure 2.4: Profil de performance basé sur le nombre d'itération

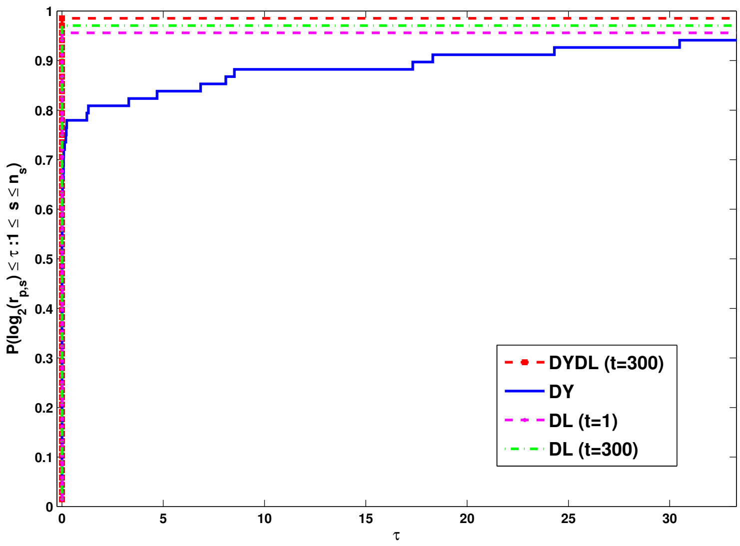


Figure 2.5: Profil de performance pour l'évaluations de gradients

Comme mentionné dans la section 2.1, nous avons développé une nouvelle méthode hybride de gradient conjugué en accord avec la méthode de Newton, cette nouvelle algorithme vise à atteindre numériquement la convergence quadratique de la méthode de Newton sans avoir besoin d'évaluer ou stoker la matrice Hessienne, tout en conservant certaines bonnes propriétés de l'algorithme de gradient conjugué. Le chapitre 5 de notre travail est consacré à la présentation d'un nouveau test numérique qui évalue si notre algorithme DYDL approche la convergence quadratique observée dans la méthode de Newton. Ce test repose sur l'analyse des rapports d'erreur entre deux itérations successives, et permet de visualiser graphiquement, comment ces rapports d'erreur indiquent la présence d'un convergence quadratique.

Chapitre 3

Combinaison convexe de deux méthodes de gradient conjugué basées sur l'algorithme de Newton

Introduction

Dans ce travail, nous nous intéressons à la minimisation d'une fonction à n variables, $n \in N^*$. Considérons le problème d'optimisation non linéaire sans contrainte,

$$\min_{x \in \mathbb{R}^n} f(x), \tag{3.1}$$

où, f: $\mathbb{R}^n \mapsto \mathbb{R}$ est une fonction non linéaire lisse, son gradient existe est bien définie par $g = \nabla f(x)$.

Les mathématiciens ont développé de nombreuses techniques numériques pour résoudre (3.1), parmi lesquelles les méthodes de la descente la plus rapide [35, 53], les méthodes de Newton [14, 11], les méthodes du gradient conjugué [1, 56] [34] et les méthodes de quasi-Newton [55, 36].

La base de toutes ces méthodes consiste, à partir d'un vecteur initial approprié $x_0 \in \mathbb{R}^n$ à générer une séquence $\{x_k\}_{k\geq 0}$, comme suit

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k \geq 0, \tag{3.2}$$

où, α_k est la taille du pas déterminée à l'aide d'une règle de recherche linéaire, et d_k est la direction de recherche qui identifie les différentes méthodes de résolution du problème (3.1).

La direction de recherche de la méthode de Newton est calculée comme suit

$$d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}, \tag{3.3}$$

où, $\nabla^2 f(x_{k+1})$ est la matrice Hessienne de f.

Lorsque le point initial est proche de la solution, la méthode de Newton offre un taux de convergence quadratique, car elle utilise l'information de la dérivée seconde pour générer la direction de recherche. Cependant, la méthode de Newton, est efficace pour les problèmes de petite taille et ne convient pas aux problèmes à grande échelle, en termes de stockage et de coût de calcul de l'évaluation de la matrice hessienne [11].

La méthode du gradient conjugué est beaucoup plus utile et pratique pour résoudre (3.1), en particulier pour les cas à grande échelle, en raison de sa simplicité et de ses faibles besoins en mémoire, car elle n'utilise que l'information de la dérivée première [43]. La méthode du gradient conjugué possède la propriété de convergence globale, ce qui lui permet de converger vers la solution optimale, quel que soit le vecteur de départ choisi. Sa direction de la recherche est donnée comme suit

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{3.4}$$

En fonction du choix du paramètre $\beta_k \in \mathbb{R}$, appelé paramètre du gradient conjugué, il existe plusieurs algorithmes de gradient conjugué différents. Dans ce qui suit, nous allons mentionner quelques formules célèbres pour ce paramètre.

$$\begin{aligned}
\beta_k^{HS} &= \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \text{(HS - Hestenes et Stiefel [31])},\\
\beta_k^{FR} &= \frac{\| g_{k+1} \|^2}{\| g_k \|^2}, \quad \text{(FR - Fletcher and Reeves [23])},\\
\beta_k^{PRP} &= \frac{g_{k+1}^T y_k}{\| g_k \|^2}, \quad \text{(PRP - Polak et Ribére [45, 46])},\\
\beta_k^{CD} &= \frac{\| g_{k+1} \|^2}{-d_k^T g_k}, \quad \text{(CD - conjugate descent [24])},\\
\beta_k^{LS} &= \frac{g_{k+1}^T y_k}{-d_k^T g_k}, \quad \text{(LS - Liu et Storey [40])},\\
\beta_k^{DY} &= \frac{\| g_{k+1} \|^2}{d_k^T y_k}, \quad \text{(DY - Dai et Yuan[15])},\\
\beta_k^{WYL} &= \frac{g_{k+1}^T (g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k)}{\| g_k \|^2}, \quad \text{(WYL- Wei, Yao et Liu [61, 33])}
\end{aligned}$$

Les algorithmes de gradient conjugué sont classés en trois grandes catégories à savoir, les méthodes classiques, les méthodes modifiées et les méthodes hybrides.
Les méthodes (HS),(FR), (PRP), (CD), (LS), (DY), sont connues comme des méthodes classiques en raison de leur simplicité.
La méthode du gradient conjugué (WYL) a été proposée par Wei [61, 33], comme une version modifiée de la méthode classique PRP, afin de l'améliorer et de la rendre plus efficace. Cette méthode présente non seulement des expériences numériques intéressantes, mais elle satisfait également à la condition de descente suffisante et présente des propriétés de convergence globale.

Les méthodes hybrides de gradient conjugué sont basées sur la combinaison des méthodes classiques ou modifiées afin de construire de nouvelles méthodes pratiques qui présentent les avantages des méthodes à combiner. Ainsi, plusieurs méthodes hybrides sont suggérées, par exemple, Andrei [6] a proposé de combiner les méthodes du gradient conjugué (DY) et (HS), sous la forme d'une combinaison convexe et a distingué cette méthode en faisant en sorte que sa direction de recherche, soit la direction newtonienne, à l'aide de l'équation de la sécante, afin d'éviter l'évaluation de la matrice de Hessien. Motivés par l'idée d'Andrei [6], récemment Fanar N. Jardow et Ghada M. Al-Naemi [37], ainsi que Djordjević [21], ont dérivé de nouvelles méthodes hybrides de gradient conjugué qui satisfont à la propriété de descente suffisante, de telle sorte que les directions de Newton sont employées.

Inspirés du travail d'Andrei [6], nous proposons un nouvel algorithme hybride de gradient conjugué, qui relie les méthodes (WYL) et (CD), en se basant sur la direction de Newton afin de bénéficier des propriétés souhaitables des méthodes de gradient conjugué et newtoniennes tout en évitant leurs inconvénients. Plus précisément, notre objectif est de préserver les caractéristiques essentielles de l'algorithme de gradient conjugué, notamment sa simplicité, sa capacité à résoudre des problèmes à grande échelle et sa propriété de convergence globale. De plus, nous visons à atteindre numériquement la convergence quadratique observée dans la méthode de Newton, tout en évitant le calcul directe de la matrice Hessienne très coûteux et la sensibilité de sa convergence au vecteur de départ sélectionné.

3.1 Algorithme de gradient conjugué hybride

Dans cette section, nous décrivons notre nouvelle méthode de gradient conjugué pour les problèmes d'optimisation sans contrainte à grande échelle, en calculant le paramètre β_k, désigné dans par β_k^{wylcd}, comme une combinaison convexe entre β_k^{CD} et β_k^{WYL}. c'est-à-dire

$$\beta_k^{wylcd} = (1 - \gamma_k)\beta_k^{WYL} + \gamma_k \beta_k^{CD}, \tag{3.5}$$

où, $\gamma_k \in [0, 1]$.

La direction d_k est donc donnée par

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + (1 - \gamma_k)\beta_k^{WYL} d_k + \gamma_k \beta_k^{CD} d_k. \tag{3.6}$$

Si $\gamma_k = 0$, alors $\beta_k^{wylcd} = \beta_k^{WYL}$ et si $\gamma_k = 1$, alors $\beta_k^{wylcd} = \beta_k^{CD}$. D'autre part, si $0 < \gamma_k < 1$ alors β_k^{wylcd} est la combinaison convexe entre β_k^{CD} et β_k^{WYL}.

Supposons que $\nabla^2 f(x_k)^{-1}$ existe $\forall k \geq 0$ pour la fonction objectif f.
Comme nous le savons, la méthode de Newton a des propriétés de convergence quadratique, nous allons donc construire une nouvelle méthode hybride de gradient conjugué, en accord avec la méthode de Newton, dans le but d'atteindre numériquement le comportement de convergence quadratique observé dans la méthode de Newton, tout en conservant certaines bonnes propriétés de l'algorithme de gradient conjugué. Pour ce faire, inspirés par le travail d'Andrei [6], nous calculons le γ_k dans (3.5) de manière à ce que notre direction de recherche, donnée par la relation (3.6), soit égale à la direction de Newton. C-à-d

$$-g_{k+1} + (1-\gamma_k)\beta_k^{WYL} d_k + \gamma_k \beta_k^{CD} d_k = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}. \tag{3.7}$$

En multipliant les deux côtés de l'équation (3.7) par $s_k^T \nabla^2 f(x_{k+1})$ à partir de la gauche, on obtient

$$-s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (1-\gamma_k)\beta_k^{WYL} s_k^T \nabla^2 f(x_{k+1}) d_k + \gamma_k \beta_k^{CD} s_k^T \nabla^2 f(x_{k+1}) d_k = -s_k^T g_{k+1},$$

En simplifiant les calcules, nous obtenant

$$\gamma_k(\beta_k^{CD} - \beta_k^{WYL}) s_k^T \nabla^2 f(x_{k+1}) d_k + s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - \beta_k^{WYL} s_k^T \nabla^2 f(x_{k+1}) d_k = -s_k^T g_{k+1}$$

ainsi

$$\gamma_k(\beta_k^{CD} - \beta_k^{WYL}) s_k^T \nabla^2 f(x_{k+1}) d_k = -s_k^T g_{k+1} + s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - \beta_k^{WYL} s_k^T \nabla^2 f(x_{k+1}) d_k$$

ce qui donne

$$\gamma_k \quad = \quad \frac{-s_k^T g_{k+1} + s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - \beta_k^{WYL} s_k^T \nabla^2 f(x_{k+1}) d_k}{(-\beta_k^{WYL} + \beta_k^{CD}) s_k^T \nabla^2 f(x_{k+1}) d_k}. \tag{3.8}$$

où, $s_k = x_{k+1} - x_k$.

Pour calculer γ_k, nous devons obtenir la matrice hessienne de la fonction objectif, mais nous savons que, pour les problèmes à grande échelle, le calcul de la matrice hessienne est soit impossible, soit coûteux en pratique. Sachant cela, pour les algorithmes quasi-Newton, la matrice d'approximation B_k de la matrice hessienne $\nabla^2 f(x_k)$ est mise à jour de manière à ce que la nouvelle matrice B_{k+1} satisfasse l'équation de la sécante

$$B_{k+1} s_k = y_k.$$

Ainsi, pour obtenir un algorithme de résolution de problèmes largement utilisé, nous supposons que la paire (s_k, y_k) satisfait l'équation sécante.

$$\nabla^2 f(x_{k+1}) s_k = y_k,$$

c'est à dire

$$s_k^T \nabla^2 f(x_{k+1}) = y_k^T. \tag{3.9}$$

En remplaçant la relation (3.9) dans (3.8), on obtient donc

$$\gamma_k \quad = \quad \frac{-s_k^T g_{k+1} + y_k^T g_{k+1} - \beta_k^{WYL} y_k^T d_k}{(-\beta_k^{WYL} + \beta_k^{CD}) y_k^T d_k}. \tag{3.10}$$

Il est clair que nous avons construit une nouvelle méthode hybride de gradient conjugué conforme à la méthode de Newton, mais le processus itératif est simple et conçu pour résoudre des problèmes à grande échelle, car nous avons évité le coût de calcul associé à l'évaluation directe de la matrice hessienne, en utilisant l'équation sécante.

Nous allons maintenant décrire notre algorithme, appelé "algorithme WYLCD" qui présente certaines bonnes caractéristiques à la fois de l'algorithme de gradient conjugué et de l'algorithme de Newton.

Algorithme WYLCD

Etape 0: Choisir le point initial $x_0 \in \mathbb{R}^n$, $\epsilon > 0$.
Calculer $f_0 = f(x_0)$ et $g_0 = \nabla f(x_0)$.
Définir $d_0 = -g_0$, l'estimation initiale $\alpha_0 = \frac{1}{\|g_0\|}$. Soit $k = 0$.

Etape 1: Tester un critère pour arrêter les itérations, c'est-à-dire si $\| g_k \| \leq \epsilon$, alors arrêter. Sinon, passer à l'étape 2.
Etape 2: Calculer la taille de pas α_k en utilisant les conditions de strong Wolfe Powel

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \tag{3.11}$$
$$|g_{k+1}^T d_k| \leq \sigma |g_k^T d_k| \tag{3.12}$$

où, $0 < \delta < \frac{1}{2}, 0 < \sigma < \frac{1}{5}$.
Etape 3: Mettre à jour la prochaine itération par $x_{k+1} = x_k + \alpha_k d_k$. Calculer $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$ et $s_k = x_{k+1} - x_k$.
Etape 4: Si $(-\beta_k^{WYL} + \beta_k^{CD}) y_k^T d_k = 0$, alors $\gamma_k = 0$, sinon calculer γ_k comme dans (3.10).
Etape 5: Si $\gamma_k \leq 0$, alors calculer $\beta_k^{wylcd} = \beta_k^{WYL}$.
Si $\gamma_k \geq 1$, alors calculer $\beta_k^{wylcd} = \beta_k^{CD}$.
Si $0 < \gamma_k < 1$, alors calculer β_k^{wylcd} comme dans (3.5).
Etape 6: Calculer $d_{k+1} = -g_{k+1} + \beta_k^{wylcd} d_k$. Définir la valeur initiale $\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$.
Etape 7: Soit $k = k + 1$, Revenir à l'étape 1.

3.2 La condition de descente suffisante

Le théoreme suivant confirme que notre nouvelle direction, donné par 3.6 satisfait la condition de descente suffisante.

Théoreme 13. .
Supposons que les séquences $\{g_k\}_{k \geq 0}$ et $\{d_k\}_{k \geq 0}$, soient générées par l'algorithme "WYLCD". Supposons également que α_k, soit déterminé par la recherche linéaire de strong Wolfe-Powell (3.11) et (3.12). Si σ dans (3.12) vérifie $0 < \sigma < \frac{1}{5}$, alors la condition de descente suffisante,

$$g_{k+1}^T d_{k+1} \leq -c \| g_{k+1} \|^2 \tag{3.13}$$

est satisfait.

Démonstration 4. .
Si $\gamma_k = 0$, alors $\beta_k^{wylcd} = \beta_k^{WYL}$ and si $\gamma_k = 1$, alors $\beta_k^{wylcd} = \beta_k^{CD}$, la condition de descente suffisante est déjà prouvée en [61, 33] et [24] respectivement.

Maintenant, nous prouvons la condition de descente suffisante dans le cas où $0 < \gamma_k < 1$. D'après la formule de β_k^{wylcd} dans (3.5), nous avons

$$|\beta_k^{wylcd}| \leq (1 - \gamma_k)|\beta_k^{WYL}| + \gamma_k |\beta_k^{CD}|$$

puisque $0 < \gamma_k < 1$, alors

$$|\beta_k^{wylcd}| \leq |\beta_k^{WYL}| + |\beta_k^{CD}| \tag{3.14}$$

En remplacant les formules des β_k^{wylcd} et β_k^{CD} dans la relation (3.14), nous obtenons

$$|\beta_k^{wylcd}| \leq \frac{\| g_{k+1} \|^2 + \frac{\|g_{k+1}\|}{\|g_k\|} |g_{k+1}^T g_k|}{\| g_k \|^2} + \frac{\| g_{k+1} \|^2}{|-g_k^T d_k|} \tag{3.15}$$

sachant que $|g_{k+1}^T g_k| < \| g_{k+1} \| \| g_k \|$, alors la relation (3.15) devient,

$$|\beta_k^{wylcd}| \leq \frac{\| g_{k+1} \|^2 + \frac{\|g_{k+1}\|}{\|g_k\|} \| g_{k+1} \| \| g_k \|}{\| g_k \|^2} + \frac{\| g_{k+1} \|^2}{|g_k^T d_k|} \tag{3.16}$$

En simplifiant et en regroupant les $\| g_{k+1} \|$, nous obtenons

$$|\beta_k^{wylcd}| \leq \frac{2 \| g_{k+1} \|^2}{\| g_k \|^2} + \frac{\| g_{k+1} \|^2}{|g_k^T d_k|}. \tag{3.17}$$

Ensuite, en multipliant l'inégalité (3.17) à partir des deux cotés par $|g_{k+1}^T d_k|$, nous obtenons

$$|\beta_k^{wyld} g_{k+1}^T d_k| \leq \frac{2 \| g_{k+1} \|^2}{\| g_k \|^2} |g_{k+1}^T d_k| + \frac{\| g_{k+1} \|^2}{|g_k^T d_k|} |g_{k+1}^T d_k|, \tag{3.18}$$

en utilisant (3.12) il en résulte

$$|\beta_k^{wylcd} g_{k+1}^T d_k| \leq \frac{2 \parallel g_{k+1} \parallel^2}{\parallel g_k \parallel^2} \sigma |g_k^T d_k| + \frac{\parallel g_{k+1} \parallel^2}{|g_k^T d_k|} \sigma |g_k^T d_k|$$

ainsi

$$|\beta_k^{wylcd} g_{k+1}^T d_k| \leq \frac{2 \parallel g_{k+1} \parallel^2}{\parallel g_k \parallel^2} \sigma |g_k^T d_k| + \sigma \parallel g_{k+1} \parallel^2, \tag{3.19}$$

En multipliant (3.6) par g_{k+1} on trouve

$$g_{k+1}^T d_{k+1} = - \parallel g_{k+1} \parallel^2 + \beta_k^{wylcd} g_{k+1}^T d_k. \tag{3.20}$$

alors

$$\frac{g_{k+1}^T d_{k+1}}{\parallel g_{k+1} \parallel^2} = -1 + \beta_k^{wylcd} \frac{g_{k+1}^T d_k}{\parallel g_{k+1} \parallel^2}. \tag{3.21}$$

Prouvons maintenant la propriété de descente de la direction d_k par induction. Nous avons $g_1^T d_1 = - \parallel g_1 \parallel^2$, nous supposons que d_i, $i = 1, 2, \ldots, k$ sont toutes des directions de descente ($g_i^T d_i < 0$). D'après (3.19), il en résulte

$$|\beta_k^{wylcd} g_{k+1}^T d_k| \leq -2\sigma \parallel g_{k+1} \parallel^2 \frac{g_k^T d_k}{\parallel g_k \parallel^2} + \sigma \parallel g_{k+1} \parallel^2 . \tag{3.22}$$

ainsi

$$2\sigma \parallel g_{k+1} \parallel^2 \frac{g_k^T d_k}{\parallel g_k \parallel^2} - \sigma \parallel g_{k+1} \parallel^2 \leq \beta_k^{wylcd} g_{k+1}^T d_k \leq -2\sigma \parallel g_{k+1} \parallel^2 \frac{g_k^T d_k}{\parallel g_k \parallel^2} + \sigma \parallel g_{k+1} \parallel^2, \tag{3.23}$$

d'après (3.21) et (3.23) nous déduisons

$$\frac{g_{k+1}^T d_{k+1}}{\parallel g_{k+1} \parallel^2} \leq -1 - 2\sigma \frac{g_k^T d_k}{\parallel g_k \parallel^2} + \sigma. \tag{3.24}$$

Répéter ce processus

$$\begin{aligned}
\frac{g_{k+1}^T d_{k+1}}{\parallel g_{k+1} \parallel^2} &\leq -1 - 2\sigma \frac{g_k^T d_k}{\parallel g_k \parallel^2} + \sigma \\
&\leq -1 - 2\sigma(-1 - 2\sigma \frac{g_{k-1}^T d_{k-1}}{\parallel g_{k-1} \parallel^2} + \sigma) + \sigma \\
&\leq -1 + 2\sigma + (2\sigma)^2(-1 - 2\sigma \frac{g_{k-2}^T d_{k-2}}{\parallel g_{k-2} \parallel^2} + \sigma) - 2\sigma^2 + \sigma \\
&\leq -1 + 2\sigma - (2\sigma)^2 - (2\sigma)^3(-1 - 2\sigma \frac{g_{k-3}^T d_{k-3}}{\parallel g_{k-3} \parallel^2} + \sigma) + \sigma(2\sigma)^2 - 2\sigma^2 + \sigma \\
&\leq -1 + 2\sigma - (2\sigma)^2 + (2\sigma)^3 + (2\sigma)^4(-1 - 2\sigma \frac{g_{k-4}^T d_{k-4}}{\parallel g_{k-4} \parallel^2} + \sigma) - \sigma(2\sigma)^3 + \sigma(2\sigma)^2 \\
&- 2\sigma^2 + \sigma \\
&\vdots \\
&\leq -1 + 2\sigma - (2\sigma)^2 + (2\sigma)^3 - (2\sigma)^4 + \cdots - (2\sigma)^{k-1}(-1 - 2\sigma \frac{g_{k-(k-1)}^T d_{k-(k-1)}}{\parallel g_{k-(k-1)} \parallel^2} + \sigma) \\
&- \sigma(2\sigma)^3 + \sigma(2\sigma)^2 - 2\sigma^2 + \sigma \\
&\leq -1 + 2\sigma - (2\sigma)^2 + (2\sigma)^3 - (2\sigma)^4 + \cdots + (2\sigma)^{k-1} + (2\sigma)^k \frac{g_1^T d_1}{\parallel g_1 \parallel^2} - \sigma(2\sigma)^{k-1} \\
&- \sigma(2\sigma)^3 + \sigma(2\sigma)^2 - 2\sigma^2 + \sigma,
\end{aligned} \tag{3.25}$$

en utilisant $g_1^T d_1 = - \parallel g_1 \parallel^2$, l'inégalité (3.25) devient

$$\begin{aligned}
\frac{g_{k+1}^T d_{k+1}}{\parallel g_{k+1} \parallel^2} &\leq -1 + [(2\sigma + (2\sigma)^3 + \cdots + (2\sigma)^{k-1}) - ((2\sigma)^2 + (2\sigma)^4 + \cdots + (2\sigma)^k)] \\
&+ [(\sigma + \sigma(2\sigma)^2 + \cdots + \sigma(2\sigma)^{k-2}) - (2\sigma^2 + \sigma(2\sigma)^3 + \cdots + \sigma(2\sigma)^{k-1})]
\end{aligned}$$

Donc

$$\begin{aligned}\frac{g_{k+1}^T d_{k+1}}{\| g_{k+1} \|^2} &\leq -1 + [(2\sigma + (2\sigma)^3 + \cdots + (2\sigma)^{k-1}) + ((2\sigma)^2 + (2\sigma)^4 + \cdots + (2\sigma)^k)] \\ &+ [(\sigma + \sigma(2\sigma)^2 + \cdots + \sigma(2\sigma)^{k-2}) + (2\sigma^2 + \sigma(2\sigma)^3 + \cdots + \sigma(2\sigma)^{k-1})] \\ &= -1 + \sum_{j=1}^{k}(2\sigma)^j + \sigma\sum_{j=0}^{k-1}(2\sigma)^j\end{aligned}$$

ainsi

$$\frac{g_{k+1}^T d_{k+1}}{\| g_{k+1} \|^2} \leq -2 + \sum_{j=0}^{k}(2\sigma)^j + \sigma\sum_{j=0}^{k-1}(2\sigma)^j, \tag{3.26}$$

il en résulte

$$\sum_{j=0}^{k}(2\sigma)^j < \sum_{j=0}^{\infty}(2\sigma)^j = \frac{1}{1-(2\sigma)} \qquad \textit{et} \qquad \sum_{j=0}^{k-1}(2\sigma)^j < \sum_{j=0}^{\infty}(2\sigma)^j = \frac{1}{1-(2\sigma)}.$$

Par conséquent, l'inégalité (3.26) *devient*

$$\frac{g_{k+1}^T d_{k+1}}{\| g_{k+1} \|^2} \leq -(2 - \frac{1+\sigma}{1-(2\sigma)}), \tag{3.27}$$

en prenant $0 < \sigma < \frac{1}{5}$*, alors*

$$g_{k+1}^T d_{k+1} \leq -(2 - \frac{1+\sigma}{1-(2\sigma)}) \| g_{k+1} \|^2 < 0 \tag{3.28}$$

Ainsi, par induction, $g_k^T d_k < 0$ *est valable pour tout* $k \geq 0$.

Nous prouvons maintenant la condition de descente suffisante de d_k*. Si* $0 < \sigma < \frac{1}{5}$*, il suffit de prendre* $c = 2 - \frac{1+\sigma}{1-(2\sigma)}$*, où* $0 < c < 1$*. Ensuite, à partir de l'inégalité* (3.28)*, on obtient la condition de descente suffisante*

$$g_{k+1}^T d_{k+1} \leq -c \| g_{k+1} \|^2, \tag{3.29}$$

3.3 Propriétés de convergence de notre méthode

Les hypothèses essentielles [7] suivantes sur la fonction objective sont nécessaires pour établir la convergence globale de notre méthode hybride (WYLCD).

H1. L'ensemble de niveaux $\mathcal{H} = \{x \in R^n / f(x) \leq f(x_0)\}$ est borné, où x_0 est le vecteur initial.

H2. Dans un certain voisinage $\mathcal{Q}$ de $\mathcal{H}$, la fonction objective f est continument différentiable, son gradient est continu Lipschitz, par conséquent $\exists l > 0$ tel que

$$\| g(x) - g(y) \| \leq l \| x - y \| \qquad \forall x, y \in \mathcal{Q}. \tag{3.30}$$

Ces hypothèses impliquent que, $\exists \bar{r} \geq 0$ tel que

$$\| g(x) \| \leq \bar{r}. \quad \forall x \in \mathcal{H} \tag{3.31}$$

Les lemmes 3 et 4 vont utilisé pour démontrer le théorème de convergence ci après.

Lemme 3. *[15].*
Supposons que les hypothèses H1 et H2 mentionnées ci-dessus soient satisfaites, et considérons les méthodes formulées par (3.2) et (3.4), où $\{d_k\}$ *est une direction de descente et* α_k *est calculé à l'aide de la recherche linéaire strong Wolfe-Powell.*
Si

$$\sum_{k\geq 0} \frac{1}{\| d_k \|} = +\infty \tag{3.32}$$

alors

$$\lim_{k\mapsto\infty} inf \| g_k \| = 0. \tag{3.33}$$

Lemme 4. *[39].*
Supposons que les hypothèses H1 et H2 ci-dessus se vérifient. Si d_k est une direction de descente et que la longueur de pas α_k satisfait

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \quad \sigma < 1, \tag{3.34}$$

alors

$$\alpha_k \geq \frac{1-\sigma}{L} \frac{|d_k^T g_k|}{\| d_k \|^2}. \tag{3.35}$$

Selon le lemme 4 et les conditions (3.12), (4.24) et (3.31) nous en déduisons que α_k obtenu dans le nouvel algorithme WYLCD n'est pas égal à zéro, ce qui signifie que $\exists p > 0$ tel que

$$\alpha_k \geq p, \quad \text{for all} \quad k \geq 0. \tag{3.36}$$

Théoreme 14. *Supposons que les hypothèses H1 et H2 se vérifie. Soit la séquence $\{x_k\}_{k\geq 0}$ générée par "l'algorithme WYLCD" proposé. Alors*

$$\lim_{k\mapsto\infty} inf \| g_k \| = 0. \tag{3.37}$$

Démonstration 5. *On va démontrer ce théorème en utilisant un raisonnement par l'absurde. Supposons que la relation (3.37) n'est pas vérifié. Alors il existe $r > 0$ tel que*

$$\| g_k \| > r. \tag{3.38}$$

En utilisant (3.38) et (3.13) nous avons

$$-g_k^T d_k \geq c \| g_k \|^2 \geq cr^2 \tag{3.39}$$

ainsi

$$\frac{1}{-g_k^T d_k} \leq \frac{1}{cr^2}. \tag{3.40}$$

Sachant que $s_k = x_{k+1} - x_k$, soit $A = \max\{\| x - y \| / x, y \in \mathcal{H}\}$ le diamètre de l'ensemble de niveau $\mathcal{H}$ c-à-d

$$\| s_k \| \leq A \tag{3.41}$$

D'après la définition de la direction d_k (dans (3.5) et (3.6)), nous avons

$$\| d_{k+1} \| \leq \| g_{k+1} \| + |\beta_k^{wylcd}| \| d_k \| . \tag{3.42}$$

En ce qui concerne la bornitude de β_k^{wylcd} , il existe trois cas. Si $\gamma_k = 0$, alors $\beta_k^{wylcd} = \beta_k^{WYL}$ et si $\gamma_k = 1$, alors $\beta_k^{wylcd} = \beta_k^{CD}$, Ces deux cas ont déjà été prouvés respectivement dans [61, 33] et [24]. D'après la formule de β_k^{wylcd} dans (3.5), nous avons

$$|\beta_k^{wylcd}| \leq (1-\gamma_k)|\beta_k^{WYL}| + \gamma_k|\beta_k^{CD}|$$

puisque $0 < \gamma_k < 1$, alors

$$|\beta_k^{wylcd}| \leq |\beta_k^{WYL}| + |\beta_k^{CD}| \tag{3.43}$$

En remplaçant les formules des β_k^{wylcd} et β_k^{CD} dans la relation (3.43), nous obtenons

$$|\beta_k^{wylcd}| \leq \frac{\| g_{k+1} \|^2 + \frac{\|g_{k+1}\|}{\|g_k\|}|g_{k+1}^T g_k|}{\| g_k \|^2} + \frac{\| g_{k+1} \|^2}{|-g_k^T d_k|} \tag{3.44}$$

sachant que $|g_{k+1}^T g_k| < \| g_{k+1} \| \| g_k \|$, alors la relation (3.44) devient,

$$|\beta_k^{wylcd}| \leq \frac{\| g_{k+1} \|^2 + \frac{\|g_{k+1}\|}{\|g_k\|} \| g_{k+1} \| \| g_k \|}{\| g_k \|^2} + \frac{\| g_{k+1} \|^2}{|g_k^T d_k|} \tag{3.45}$$

En simplifiant et en regroupant les $\| g_{k+1} \|$, nous obtenons

$$|\beta_k^{wylcd}| \leq \frac{2 \| g_{k+1} \|^2}{\| g_k \|^2} + \frac{\| g_{k+1} \|^2}{|g_k^T d_k|}. \tag{3.46}$$

En utilisant (3.38), (4.77) et (3.31) la relation (3.46) devient

$$|\beta_k^{wylcd}| \quad \leq \tfrac{2\bar{r}^2}{r^2} + \tfrac{\bar{r}^2}{cr^2} = E, \tag{3.47}$$

avec, E est un constant positive.
D'après (3.47) et (3.31), la relation (3.42) devient

$$\| d_{k+1} \| \leq \bar{r} + E \| d_k \| . \tag{3.48}$$

Sachant que $s_k = \alpha_k d_k$, utilisant $\| d_k \| = \frac{\|s_k\|}{\alpha_k}$ et à partir de (3.36) et (3.41), nous obtenons

$$\| d_{k+1} \| \leq \bar{r} + E \frac{\| s_k \|}{\alpha_k} \leq \bar{r} + E \frac{A}{p} = M,$$

avec M est un constant positive.
Ce qui donne

$$\frac{1}{\| d_{k+1} \|} \geq \frac{1}{M}.$$

Donc

$$\sum_{k\geq 0} \frac{1}{\| d_{k+1} \|} \geq \sum_{k\geq 0} \frac{1}{M}.$$

Par conséquent

$$\sum_{k\geq 0} \frac{1}{\| d_k \|} = +\infty. \tag{3.49}$$

En appliquant le lemme 3, on conclut que

$$\lim_{k \mapsto \infty} inf \| g_k \| = 0$$

c'est une contradiction avec (3.38), donc nous avons prouvé (3.37).

3.4 Les expériences numériques

Dans cette section, nous allons décrire les résultats numériques du nouvel algorithme proposé (algorithme WYLCD). Pour évaluer l'efficacité de notre algorithme "WYLCD", nous l'avons testé par rapport aux algorithmes (WYL) [61, 33] et (CD) [24] à partir desquels il a été construit, en utilisant certains problèmes de test choisis dans [8]. Pour chaque fonction, nous avons réalisé des expériences numériques pour le nombre de variables 2 . . . 10 000. Dans les expériences numériques suivantes, tous les codes sont compilés sur un PC ayant les spécifications suivantes :Intel(R) Core(TM) i5-3210M CPU 2.50GHz 2.50 GHz, 4.00 Go RAM, en utilisant le profil Dolan et Moré [22] comme outil d'évaluation. Tous les algorithmes utilisent les conditions de recherche de la ligne de Wolfe Powel forte avec les paramètres $\delta = 0.0001$ et $\sigma = 0.1$ et se terminent lorsque $\| g_k \|_\infty \leq 10^{-6}$.
Les figures 3.1, 3.2 et 3.3 représentent les performances de ces algorithmes en fonction des indicateurs suivants : temps CPU, nombre d'itérations, évaluations du gradient. Comme le montrent toutes les figures ci-dessus, le nouvel algorithme WYLCD est clairement supérieur aux autres algorithmes.

Comme nous l'avons mentionné dans la section 3.1, nous avons construit un nouvel algorithme hybride de gradient conjugué basé sur la direction de Newton, dans le but d'approcher la convergence quadratique de la méthode de Newton du point de vue numérique et d'obtenir certaines caractéristiques essentielles de l'algorithme de gradient conjugué. Dans le chapitre 5, nous proposons un nouveau test numérique basé sur certaines problèmes de test choisies dans [8] avec différentes dimensions, pour étudier la convergence quadratique de notre méthode.

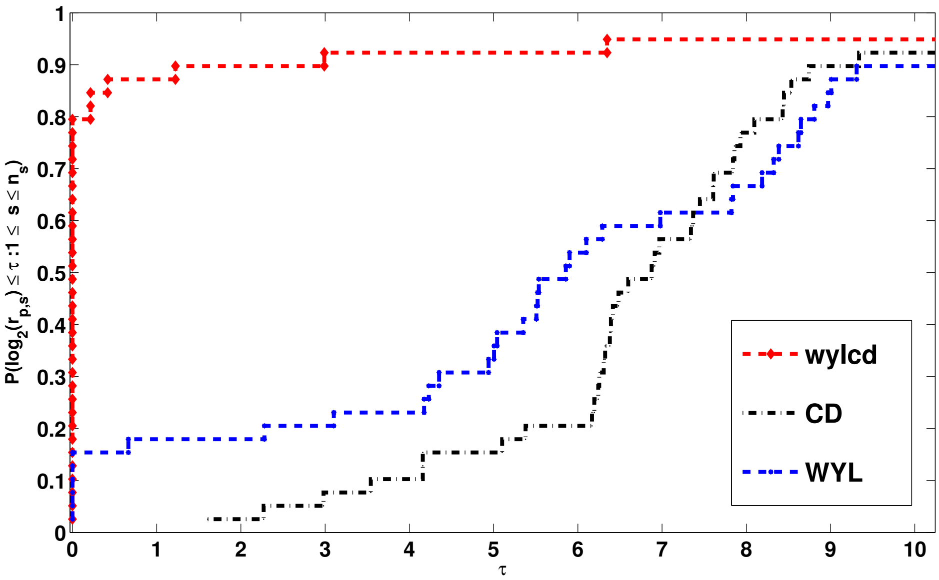


Figure 3.1: Profil de performance pour le temps CPU

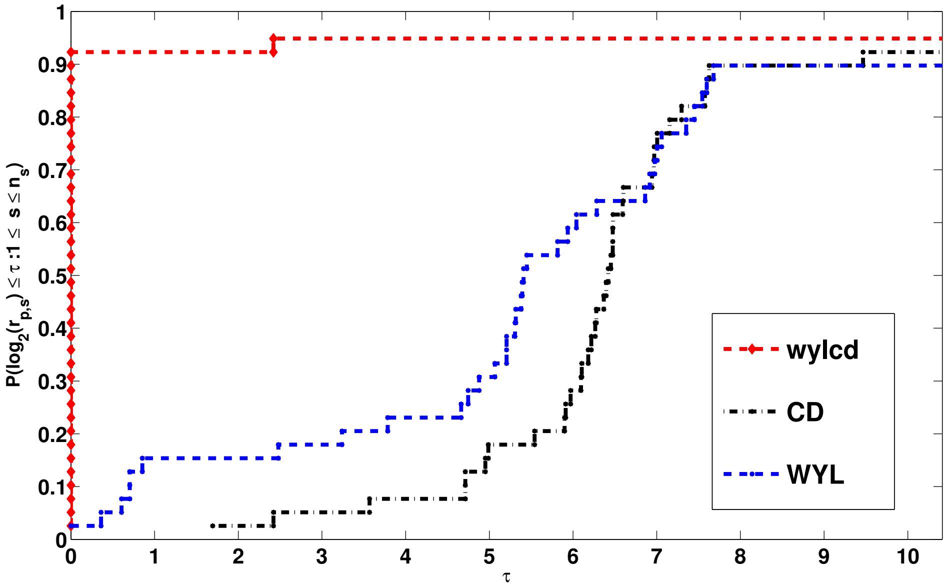


Figure 3.2: Profil de performance pour le nombre d'itération

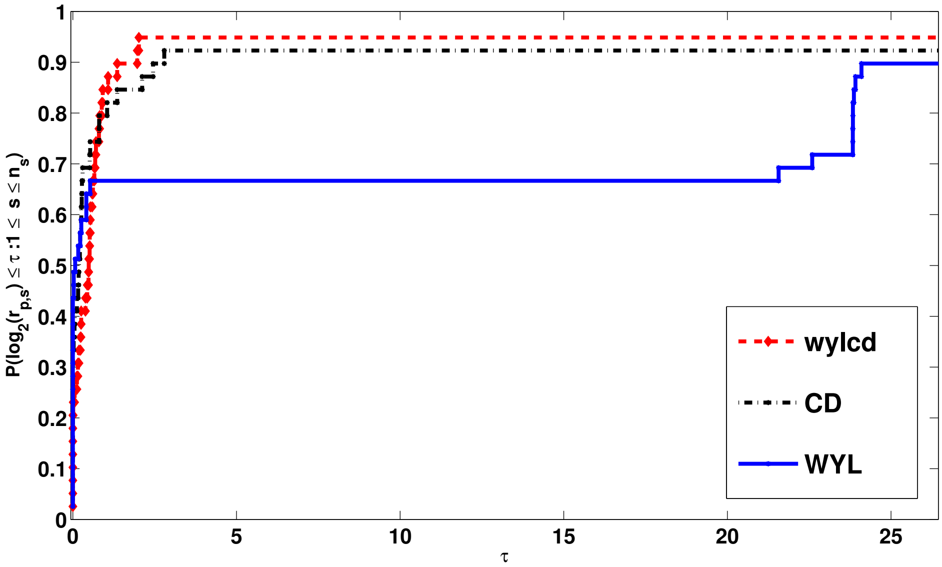


Figure 3.3: Profil de performance pour l'évaluations de gradients

Chapitre 4

La relation entre trois directions du gradient conjugué et la direction de Newton.

Introduction

Nous nous intéressons aux problèmes d'optimisation non linéaire sans contrainte, formulés comme suit

$$\min_{x \in \mathbb{R}^n} f(x), \tag{4.1}$$

où, f: $\mathbb{R}^n \mapsto \mathbb{R}$ est deux fois continûment différentiable; son gradient et sa matrice hessienne sont respectivement notés $g(x)$ et $\nabla^2 f(x)$. De nombreuses méthodes de recherche linéaire ont été développées pour résoudre (4.1), qui servent toutes à générer une séquence $\{x_k\}_{k\geq 0}$, comme suit :

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k, \tag{4.2}$$

où, α_k est la taille du pas et d_k est la direction de recherche.
La taille du pas α_k est principalement choisie, pour satisfaire des conditions de recherche linéaire inexactes. Ici, nous considérons les conditions de la règle de Wolfe forte [51, 52].

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \tag{4.3}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{4.4}$$

où, $0 < \delta < \sigma < \frac{1}{2}$.
Les différents choix de d_k, identifient les différentes méthodes pour résoudre (4.1).

Si nous choisissons

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k d_k, \quad \beta_k \in \mathbb{R} \tag{4.5}$$

comme direction de recherche à chaque itération, la méthode correspondante qui génère la séquence (4.2) est appelée méthode du gradient conjugué. La méthode du gradient conjugué est très utile et pratique pour résoudre (4.1) en raison de sa simplicité, de ses besoins de stockage réduits et de sa pertinence pour une utilisation avec de grandes valeurs de n [43]. La méthode du gradient conjugué possède la propriété de convergence globale, c'est-à-dire qu'elle converge vers la solution quel que soit le vecteur initial [43]. Le scalaire β_k appelé paramètre du gradient conjugué, détermine les différentes méthodes du gradient conjugué. Dans ce qui suit, nous donnons quelques valeurs célèbres de β_k.

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2},$$

$$\beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k}, \quad \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \quad \beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k},$$

où, $y_k = g_{k+1} - g_k$, $s_k = \alpha_k d_k$ et $t > 0$.
Les méthodes correspondantes sont appelées Hestenes-Stiefel [31], méthode de Fletcher-Reeves (FR) [23], méthode de Polak-Ribiére-Polayk (PRP) [45, 46], méthode de descente conjuguée [24], méthode de Dai-Yuan (DY) [15], méthode de Dai-Liao (DL) [16], respectivement.

De nombreux chercheurs ont entrepris de développer de nouvelles méthodes hybrides du gradient conjugué en combinant des approches classiques. Ces méthodes hybrides ont été largement confirmées comme étant efficaces par de nombreuses expériences numériques. Par exemple,

$$\beta_k = (1-\theta_k)\beta_k^{DL} + \theta_k \beta_k^{DY}, \quad 0 < \theta_k < 1 \tag{4.6}$$

$$d_{k+1} = \delta_k d_k^{FR} + \gamma_k d_k^{PRP} + (1-\delta_k-\gamma_k) d_k^{DY}, \quad 0 < \delta_k, \gamma_k < 1 \tag{4.7}$$

$$d_{k+1} = \lambda_k d_k^{DY} + \theta_k d_k^{CD} + (1-\lambda_k-\theta_k) d_k^{HS}, \quad 0 < \lambda_k, \theta_k < 1 \tag{4.8}$$

Les méthodes correspondantes : Hamel et al. [26], Ben Hanachi et al. [28], Hallal et al. [29] respectivement.

Si nous sélectionnons

$$d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}, \tag{4.9}$$

comme direction de recherche à chaque itération, la méthode correspondante qui génère la séquence (4.2) est la méthode de Newton. Cette méthode se caractérise par une convergence quadratique lorsque l'initialisation est proche de la solution [11]. La méthode de Newton est efficace pour les problèmes de petite taille, mais elle n'est pas adaptée aux problèmes à grande taille en raison des besoins de stockage et des coûts de calcul élevés pour évaluer la matrice hessienne [11].

L'efficacité de la méthode de Newton pour les fonctions non convexes dépend du vecteur initial sélectionné et peut être divergente, s'il est trop éloigné de la solution [11]. De plus, de nombreux efforts ont été déployés pour améliorer la méthode de Newton, grâce à des algorithmes pratiques, comme la méthode quasi-Newton, qui remplace le calcul coûteux de la matrice hessienne par une matrice approximative, facilement mise à jour [36]. Cependant, ces améliorations impliquent toujours l'évaluation de matrices, ce qui les rend plus coûteuses que les algorithmes basés sur les calculs de vecteurs. C'est pourquoi nous envisageons de développer un nouvel algorithme d'optimisation qui se base uniquement sur des calculs de vecteurs, converge quel que soit le vecteur initial et est conçu pour construire une direction qui approxime étroitement la direction de Newton.

Dans [5], Andrei a introduit une méthode hybride du gradient conjugué en tant que combinaison convexe de deux directions de descente d_k^{DY} et d_k^{HS}. Le paramètre d'hybridation est calculé de manière à ce que la nouvelle direction soit égale à la direction de Newton.

$$-g_{k+1} + (1-\theta_k)\beta_k^{HS} d_k + \theta_k \beta_k^{DY} d_k = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}. \tag{4.10}$$

En multipliant les deux côtés de l'équation ci-dessus par $s_k^T \nabla^2 f(x_{k+1})$, nous obtenons

$$-s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (1-\theta_k)\beta_k^{HS} s_k^T \nabla^2 f(x_{k+1}) d_k + \theta_k \beta_k^{DY} s_k^T \nabla^2 f(x_{k+1}) d_k = -s_k^T g_{k+1}.$$

où $0 < \theta_k < 1$.
Sachant que, pour l'algorithme quasi-Newton, une matrice d'approximation B_k de la hessienne $\nabla^2 f(x_k)$ est mise à jour, de sorte que la nouvelle matrice B_{k+1} satisfasse l'équation sécante $B_{k+1} s_k = y_k$. Par conséquent, afin d'obtenir un algorithme d'optimisation à grande taille, Andrei [5] évite d'évaluer la matrice hessienne en se référant à Zhang [59], qui a prouvé que y_k est une approximation de $\nabla^2 f(x_{k+1}) s_k$. Ensuite, en obtenant θ_k comme

$$\theta_k = -\frac{s_k^T g_{k+1}}{g_{k+1}^T g_k}$$

Par conséquent, la direction est une approximation de la direction de Newton. Pour plus de détails, voir [5].

Dans ce travail, motivé par les travaux d'Andrei [5], nous cherchons à développer un nouvel algorithme hybride de gradient conjugué, tel que la nouvelle direction soit construite pour approximer la direction de Newton. Pour cela, comme dans [28, 29], nous envisageons la possibilité de combiner trois vecteurs de descente de l'algorithme du gradient conjugué, sous la forme suivante:

$$d_{k+1} = \varphi_k d_k^{DY} + \lambda_k d_k^{HS} + (1-\varphi_k-\lambda_k) d_k^{DL}, \tag{4.11}$$

où φ_k et λ_k sont les paramètres d'hybridation.
Dans (4.7) et (4.8), les auteurs [28, 29], ont développé une combinaison convexe de la méthode du gradient conjugué, en prenant les paramètres d'hybridation $0 < \delta_k, \gamma_k < 1$ et en les calculant de manière à ce que la direction satisfasse la condition de conjugaison $d_{k+1}^T y_k = 0$. Dans ce travail, nous considérons les paramètres d'hybridation $\varphi_k, \lambda_k \in \mathbb{R}$ et les calculons de manière à ce que la direction proposée, approxime étroitement la direction de Newton en utilisant l'équation sécante, tout en prenant en compte le fait que ces paramètres doivent être choisis, de manière à rester bornés pour assurer la convergence de l'algorithme. Nous appelons cette méthode d'hybridation une combinaison quasi convexe de la méthode du gradient conjugué. Une discussion détaillée sera présentée dans la section 4.1.

4.1 Hybridation quasi convexe des méthodes de gradient conjugué.

Dans ce travail, nous introduisons une nouvelle méthode hybride de gradient conjugué basée sur trois vecteurs de descente d_k^{DY}, d_k^{HS} et d_k^{DL}, en définissant la direction de recherche comme suit,

$$d_0 = -g_0, \quad d_{k+1} = \varphi_k d_k^{DY} + \lambda_k d_k^{HS} + (1 - \varphi_k - \lambda_k) d_k^{DL}. \tag{4.12}$$

où, les paramètres λ_k, $\varphi_k \in \mathbb{R}$. Nous appelons cette méthode d'hybridation une combinaison quasi convexe de méthodes de gradient conjugué.
Nous pouvons écrire

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k^{QCC} d_k, \tag{4.13}$$

Nous désignons le paramètre du gradient conjugué par β_k^{QCC} qui prend la valeur de

$$\beta_k^{QCC} = \varphi_k \beta_k^{DY} + \lambda_k \beta_k^{HS} + (1 - \varphi_k - \lambda_k) \beta_k^{DL}. \tag{4.14}$$

Supposons que $\nabla^2 f(x)^{-1}$ existe à chaque point itératif pour la fonction objective f, motivé par Andrei [5], nous calculons le scalaire φ_k de sorte que notre direction de recherche donnée par (4.13), (4.14), soit égale à la direction de Newton, c'est-à-dire

$$-g_{k+1} + (\varphi_k \beta_k^{DY} + \lambda_k \beta_k^{HS} + (1 - \varphi_k - \lambda_k) \beta_k^{DL}) d_k = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}. \tag{4.15}$$

En multipliant les deux côtés de l'équation (4.15) par $s_k^T \nabla^2 f(x_{k+1})$ à gauche, nous obtenons

$$-s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (\varphi_k \beta_k^{DY} + \lambda_k \beta_k^{HS} + (1 - \varphi_k - \lambda_k) \beta_k^{DL}) s_k^T \nabla^2 f(x_{k+1}) d_k = -s_k^T g_{k+1}. \tag{4.16}$$

Nous remarquons que dans (4.16) la présence de la matrice hessienne, ce qui pose problème lorsque la dimension est grande. Que pouvons-nous faire pour éviter le calcul coûteux de la matrice hessienne ?

Sachant que, pour les algorithmes quasi-Newton, une matrice d'approximation B_k de la hessienne $\nabla^2 f(x_k)$, est mise à jour de sorte que la nouvelle matrice B_{k+1} satisfasse l'équation sécante $B_{k+1} s_k = y_k$. Rappelons que, dans [59], Zhang a montré que, pour les fonctions linéaires et quadratiques, y_k est exactement $\nabla^2 f(x_{k+1}) s_k$, c'est-à-dire $\nabla^2 f(x_{k+1}) s_k = y_k$, et pour les fonctions générales, si $\| s_k \| \mapsto 0$, alors $s_k^T \nabla^2 f(x_{k+1}) s_k - s_k^T y_k = o(\| s_k \|^3)$.
Ainsi, y_k est une approximation de $\nabla^2 f(x_{k+1}) s_k$, c'est-à-dire

$$\nabla^2 f(x_{k+1}) s_k \approx y_k, \tag{4.17}$$

donc

$$s_k^T \nabla^2 f(x_{k+1}) \approx y_k^T. \tag{4.18}$$

Par conséquent, si nous substituons (4.18) dans (4.16), la nouvelle direction (4.13), (4.14) sera une approximation de la direction de Newton.
Ainsi, à partir de (4.18), (4.16) devient

$$-y_k^T g_{k+1} + (\varphi_k \beta_k^{DY} + \lambda_k \beta_k^{HS} + (1 - \varphi_k - \lambda_k) \beta_k^{DL}) y_k^T d_k = -s_k^T g_{k+1}, \tag{4.19}$$

Substituons les valeurs de β_k^{DY}, β_k^{HS} et β_k^{DL} (prendre $t = 1$ dans β_k^{DL} pour simplification). Nous obtenons φ_k

$$\varphi_k = \frac{y_k^T g_{k+1} - \lambda_k g_{k+1}^T y_k - (1 - \lambda_k)(g_{k+1}^T y_k - g_{k+1}^T s_k) - s_k^T g_{k+1}}{(\| g_{k+1} \|^2 - g_{k+1}^T (g_{k+1} - g_k) + g_{k+1}^T s_k}, \tag{4.20}$$

alors

$$\varphi_k = -\frac{\lambda_k g_{k+1}^T s_k}{g_{k+1}^T s_k + g_k^T g_{k+1}}, \qquad \forall \lambda_k \in \mathbb{R}, \quad g_{k+1}^T s_k + g_k^T g_{k+1} \neq 0 \tag{4.21}$$

Remarque 8. *Dans ce travail, nous prendrons* $t = 1$ *dans* β_k^{DL} *pour simplifier les calculs.*

Algorithme QCC

Etape 0: Choisir le point initial $x_0 \in \mathbb{R}^n$, $\epsilon > 0, t = 1$.
Calculer $f_0 = f(x_0)$ et $g_0 = \nabla f(x_0)$.
Définir $d_0 = -g_0$, l'estimation initiale $\alpha_0 = \frac{1}{\|g_0\|}$. Soit $k = 0$.
Etape 1: Tester un critère pour arrêter les itérations, c'est-à-dire si $\| g_k \| \leq \epsilon$, alors arrêter. Sinon, passer à l'étape 2.
Etape 2: Calculer la taille de pas α_k en utilisant les conditions de strong Wolfe Powel
Etape 3: Mettre à jour la prochaine itération par $x_{k+1} = x_k + \alpha_k d_k$. Calculer $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$ et $s_k = x_{k+1} - x_k$.
Etape 4: Calculer (λ_k, φ_k) comme dans (4.66).
Etape 5: Calculer β_k^{QCC} comme dans (4.14).
Etape 6: Calculer $d = -g_{k+1} + \beta_k^{QCC} d_k$.
Etape 7: Si

$$|g_{k+1}^T g_k| \geq 0.2 \| g_{k+1} \|^2, \tag{4.22}$$

alors $d_{k+1} = -g_{k+1}$.
Sinon définir $d_{k+1} = d$.
Etape 8: Définir la valeur initiale $\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$.
Etape 9: Soit $k = k + 1$, Revenir à l'étape 1.

À partir de l'algorithme QCC, nous savons que si le critère de redémarrage de la condition de Powell [47] (4.22) n'est pas satisfait, c'est-à-dire

$$|g_{k+1}^T g_k| < 0.2 \| g_{k+1} \|^2 \tag{4.23}$$

alors nous définissons une nouvelle direction de recherche par (4.13).

4.2 La condition de descente suffisante

Le théorème suivant confirment que la condition de descente et de descente suffisante est satisfaite.

Théoreme 15. .
Soit φ_k *donné par* (4.21) *et* λ_k *supposé être choisi strictement positif* $(\lambda_k > 0)$.
Supposons que α_k *dans l'algorithme QCC, soit déterminé par les conditions de la règle de Wolfe forte* (4.3), (4.4).
si $0 < \sigma < \frac{5}{11}$, *alors la condition de descente suffisante*

$$g_{k+1}^T d_{k+1} \leq -c \| g_{k+1} \|^2 \tag{4.24}$$

est satisfaite.

Démonstration 6. .
La preuve est réalisée par induction mathématique.
Pour $d_0 = -g_0$, *on obtient,* $g_0^T d_0 = - \| g_0 \|^2 < 0$.
Si (4.22) est vérifié, $g_{k+1}^T d_{k+1} = - \| g_{k+1} \|^2 < 0$.
Si (4.22) n'est pas vérifié, alors nous obtenons (4.23) et d_k *comme dans* (4.13), (4.14).
Supposons que, $g_k^T d_k < 0$ *soit satisfait, montrons qu'il est satisfait pour* $k = k + 1$.
Multiplions la direction d_{k+1} *donnée par* (4.13), (4.14) *par* g_{k+1}, *nous obtenons*

$$g_{k+1}^T d_{k+1} = - \| g_{k+1} \|^2 + (\varphi_k \beta_k^{DY} + \lambda_k \beta_k^{HS} + (1 - \varphi_k - \lambda_k) \beta_k^{DL}) g_{k+1}^T d_k$$

donc

$$g_{k+1}^T d_{k+1} = - \| g_{k+1} \|^2 + \varphi_k \beta_k^{DY} g_{k+1}^T d_k + (1 - \varphi_k) \beta_k^{DL} g_{k+1}^T d_k + \lambda_k (\beta_k^{HS} - \beta_k^{DL}) g_{k+1}^T d_k$$

En substituant les formules de β_k^{DY}, β_k^{DL} *et* β_k^{HS} *dans la relation ci-dessus, nous obtenons*

$$\begin{aligned} g_{k+1}^T d_{k+1} &= - \| g_{k+1} \|^2 + \varphi_k \frac{\| g_{k+1} \|^2}{d_k^T y_k} g_{k+1}^T d_k + (1 - \varphi_k) \frac{g_{k+1}^T (y_k - s_k)}{d_k^T y_k} g_{k+1}^T d_k \\ &+ \lambda_k \left(\frac{g_{k+1}^T y_k}{d_k^T y_k} - \frac{g_{k+1}^T (y_k - s_k)}{d_k^T y_k} \right) g_{k+1}^T d_k, \end{aligned} \tag{4.25}$$

sachant que $y_k = g_{k+1} - g_k$, la relation (4.25) devient

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \varphi_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k + (1-\varphi_k) \frac{g_{k+1}^T((g_{k+1} - g_k) - s_k)}{d_k^T y_k} g_{k+1}^T d_k + \lambda_k \frac{g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k, \quad (4.26)$$

En simplifiant les calculs, nous arrivons à

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \varphi_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k + \frac{\parallel g_{k+1} \parallel^2 - g_{k+1}^T g_k - g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k - \varphi_k \frac{\parallel g_{k+1} \parallel^2 - g_{k+1}^T g_k - g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \lambda_k \frac{g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k, \quad (4.27)$$

En poursuivant les simplifications, nous obtenons

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \varphi_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k + \frac{\parallel g_{k+1} \parallel^2 - g_{k+1}^T g_k - g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k - \varphi_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \varphi_k \frac{-g_{k+1}^T g_k - g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \lambda_k \frac{g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k, \quad (4.28)$$

ainsi

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{g_{k+1}^T g_k + g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \varphi_k \frac{g_{k+1}^T g_k + g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \lambda_k \frac{g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k. \quad (4.29)$$

En remplaçant la formule de φ_k donné par (4.21) *dans (4.29), nous obtenons*

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{g_{k+1}^T g_k + g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k - \lambda_k \frac{g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \lambda_k \frac{g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k. \quad (4.30)$$

À partir de (4.21)*, nous avons*

$$-\frac{g_{k+1}^T g_k + g_{k+1}^T s_k}{\lambda_k s_k^T g_{k+1}} = \frac{1}{\varphi_k},$$

alors

$$g_{k+1}^T g_k + g_{k+1}^T s_k = -\frac{\lambda_k s_k^T g_{k+1}}{\varphi_k} \quad (4.31)$$

En remplaçant la relation (4.31) dans (4.30)*, il en résulte*

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k + \frac{\lambda_k g_{k+1}^T s_k}{\varphi_k d_k^T y_k} g_{k+1}^T d_k. \quad (4.32)$$

Sachant que $s_k = \alpha_k d_k$, alors

$$g_{k+1}^T d_{k+1} = -\parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k + \frac{\lambda_k \alpha_k (g_{k+1}^T d_k)^2}{\varphi_k d_k^T y_k}, \quad (4.33)$$

D'une part, en utilisant la deuxième condition de la règle de Wolfe forte (4.4)*, nous obtenons*

$$d_k^T y_k = g(x_k + \alpha_k d_k)^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k, \quad (4.34)$$

Puisque $g_k^T d_k < 0$ et $0 < \sigma < \frac{5}{11}$, alors

$$d_k^T y_k \geq -(1-\sigma) g_k^T d_k > 0, \quad (4.35)$$

D'autre part, à partir de (4.21)*, nous avons*

$$\frac{1}{\varphi_k} = -\frac{g_{k+1}^T g_k + g_{k+1}^T s_k}{\lambda_k s_k^T g_{k+1}},$$

ainsi

$$\frac{1}{\varphi_k} = -\frac{g_{k+1}^T g_k}{\lambda_k s_k^T g_{k+1}} - \frac{1}{\lambda_k}. \tag{4.36}$$

Puisque λ_k est supposé être choisi positif $\lambda_k > 0$, alors

$$\frac{1}{\varphi_k} < -\frac{g_{k+1}^T g_k}{\lambda_k s_k^T g_{k+1}}. \tag{4.37}$$

Puisque $d_k^T y_k > 0$, $\alpha_k > 0$ et $\lambda_k > 0$ alors

$$\frac{\lambda_k g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k = \frac{\lambda_k \alpha_k (g_{k+1}^T d_k)^2}{d_k^T y_k} > 0 \tag{4.38}$$

Par conséquent, en multipliant (4.37) par la relation positive $\frac{\lambda_k g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k$, nous obtenons

$$\frac{\lambda_k g_{k+1}^T s_k}{\varphi_k d_k^T y_k} g_{k+1}^T d_k < -\frac{(g_{k+1}^T g_k)(g_{k+1}^T d_k)}{d_k^T y_k}, \tag{4.39}$$

À partir de (4.39), alors (4.32) devient

$$g_{k+1}^T d_{k+1} < -\| g_{k+1} \|^2 + \frac{\| g_{k+1} \|^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(g_{k+1}^T g_k)(g_{k+1}^T d_k)}{d_k^T y_k}.$$

donc

$$g_{k+1}^T d_{k+1} < -\| g_{k+1} \|^2 + (\| g_{k+1} \|^2 - g_{k+1}^T g_k) \frac{g_{k+1}^T d_k}{d_k^T y_k}$$

En utilisant la valeur absolue, nous obtenons

$$g_{k+1}^T d_{k+1} \leq -\| g_{k+1} \|^2 + | \, \| g_{k+1} \|^2 - g_{k+1}^T g_k | \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|}$$

ainsi

$$g_{k+1}^T d_{k+1} \leq -\| g_{k+1} \|^2 + (\| g_{k+1} \|^2 + |g_{k+1}^T g_k|) \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \tag{4.40}$$

À partir de (4.23), il en résulte

$$g_{k+1}^T d_{k+1} \leq -\| g_{k+1} \|^2 + (\| g_{k+1} \|^2 + 0.2 \| g_{k+1} \|^2) \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|}$$

ainsi

$$g_{k+1}^T d_{k+1} \leq -\| g_{k+1} \|^2 + 1.2 \| g_{k+1} \|^2 \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \tag{4.41}$$

D'après la deuxième condition de la règle de Wolfe forte (4.4), cela est vérifié.

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k \tag{4.42}$$

i.e.

$$\sigma g_k^T d_k \leq g_{k+1}^T d_k \leq -\sigma g_k^T d_k.$$

Alors

$$d_k^T y_k = g_{k+1}^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k > 0$$

ainsi

$$\frac{1}{d_k^T y_k} \leq \frac{1}{-(1-\sigma) g_k^T d_k}. \tag{4.43}$$

À partir de (4.42) et (4.43), on obtient

$$\frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \leq \frac{\sigma}{1-\sigma}.$$

Alors (4.41) devient

$$g_{k+1}^T d_{k+1} \quad \leq \quad - \parallel g_{k+1} \parallel^2 + \frac{1.2\sigma}{1-\sigma} \parallel g_{k+1} \parallel^2$$

par conséquent

$$g_{k+1}^T d_{k+1} \quad \leq \quad -(1 - \frac{1.2\sigma}{1-\sigma}) \parallel g_{k+1} \parallel^2 . \tag{4.44}$$

Si $0 < \sigma < \frac{5}{11}$*, il suffit de prendre* $c = 1 - \frac{1.2\sigma}{1-\sigma} > 0$.
Ainsi (4.24) est vérifiée.

Notons que en premier temps on a considéré $\lambda_k \in \mathbb{R}$, puis dans le théorème 15 on a trouvé que λ_k doit être choisi strictement positif ($\forall \lambda_k > 0$) pour assurer que la condition de descente suffisante est vérifiée. Maintenant, dans le but d'assurer que notre direction d_k soit borné, nous allons choisi λ_k en tenant compte du fait que ces deux paramètres λ_k, φ_k doivent être facile à borner pour garantir la convergence globale de notre algorithme (une discussion détaillée sur la bornitude de λ_k, φ_k sera décrite dans la section 4.3). Ainsi, nous proposons cette valeur pour λ_k.

$$\lambda_k = |g_{k+1}^T s_k + g_k^T g_{k+1}|, \tag{4.45}$$

Il est clair que dans le cas de $g_{k+1}^T s_k + g_k^T g_{k+1} \neq 0$, alors λ_k est toujours strictement positif. De plus, selon ce choix on peut facilement simplifier la formule de φ_k ce qui permet de faciliter la bornitude de notre direction.
Ainsi, le couple (λ_k, φ_k) est donné comme suit.

$$(\lambda_k, \varphi_k) = (|g_{k+1}^T s_k + g_k^T g_{k+1}|, -\frac{\lambda_k g_{k+1}^T s_k}{g_{k+1}^T s_k + g_k^T g_{k+1}}) \tag{4.46}$$

La discussion précédente concerne le cas où ($g_{k+1}^T s_k + g_k^T g_{k+1} \neq 0$), qui est le dénominateur de φ_k. Que se passe-t-il si $g_{k+1}^T s_k + g_k^T g_{k+1} = 0$?
Dans ce cas, φ_k n'est pas bien défini, car le dénominateur est égal à 0, nous allons donc apporter une correction à ce dénominateur.
Mettons

$$m_k = g_{k+1}^T s_k + g_k^T g_{k+1} = 0 \tag{4.47}$$

Nous proposons la correction suivante pour le dénominateur m_k.

$$m_k^* = m_k + a u_k^T v_k. \tag{4.48}$$

Ainsi, le φ_k corrigé est

$$\varphi_k = -\frac{\lambda_k g_{k+1}^T s_k}{m_k^*}, \quad \lambda_k \in \mathbb{R} \tag{4.49}$$

où, $a \in \mathbb{R}^*; u_k, v_k \in \mathbb{R}^n$ doivent être déterminés de manière à ce que $a u_k^T v_k$ soit toujours différent de zéro ($a u_k^T v_k \neq 0, \quad \forall k \geq 0$). Cette technique est appelée correction vectorielle de rang un. (Ce type de technique a également été introduit pour déterminer la matrice Hessienne approximative, en utilisant une correction de matrice de rang un qui satisfait l'équation quasi-Newton (4.17), voir [49]
Maintenant, nous allons essayer de choisir $a \in \mathbb{R}^*$, $u_k, v_k \in \mathbb{R}^n$ de telle sorte que $a u_k^T v_k \neq 0, \forall k \geq 0$.
Supposons que

$$g_{k+1}^T s_k + g_k^T g_{k+1} = 0 \tag{4.50}$$

i.e.

$$g_k^T g_{k+1} = -g_{k+1}^T s_k \tag{4.51}$$

nous avons

$$|g_{k+1}^T s_k + g_k^T g_{k+1}| \quad \leq \quad |g_{k+1}^T s_k| + |g_k^T g_{k+1}|, \tag{4.52}$$

À partir de (4.51), alors (4.52) devient

$$\begin{aligned} |g_{k+1}^T s_k + g_k^T g_{k+1}| &\leq |g_{k+1}^T s_k| + |-g_{k+1}^T s_k| \\ &= 2|g_{k+1}^T s_k|, \end{aligned} \quad (4.53)$$

À partir de la deuxième condition de la règle de Wolfe forte (4.4), il en résulte

$$|g_{k+1}^T s_k + g_k^T g_{k+1}| \quad < \quad -2\sigma g_k^T s_k, \quad (4.54)$$

À partir de (4.50) et (4.54), nous avons.

$$-2\sigma g_k^T s_k > |g_{k+1}^T s_k + g_k^T g_{k+1}| = 0.$$

Alors

$$-2\sigma g_k^T s_k > 0. \quad (4.55)$$

Nous pouvons prendre, $a = -2\sigma$, $u_k^T = g_k^T$ et $v_k = s_k$ où, $0 < \sigma < \frac{5}{11}$. Par conséquent,

$$a u_k^T v_k = -2\sigma g_k^T s_k \neq 0, \quad \forall k \geq 0.$$

Donc, le dénominateur corrigé de φ_k est :

$$m_k^* = -2\sigma g_k^T s_k.$$

Alors, φ_k devient

$$\varphi_k = \frac{\lambda_k g_{k+1}^T s_k}{2\sigma g_k^T s_k}, \quad \lambda_k \in \mathbb{R}. \quad (4.56)$$

Rappelons que, dan le théorème 15, on a démontré la condition de descente sans correction, i.e. dans le cas où le dénominateur de φ_k est supposé différent de zéro. Maintenant, dans le cas où le dénominateur de φ_k est supposé égale à zéro, on a met une correction sur ce dénominateur, donc on doit encore revérifier la condition de descente, pour cela on va construire un nouveau théorème qui assure la condition de descente avec correction.

Théoreme 16. .
Supposons que, $g_{k+1}^T s_k + g_k^T g_{k+1} = 0$ et α_k dans l'Algorithme QCC est déterminé par les conditions de la règle de Wolfe forte (4.3), (4.4).
Si φ_k est donné par (4.56) et

$$|\lambda_k| < \frac{1 - 2\sigma}{0.2\sigma} \quad (4.57)$$

où, $0 < \sigma < \frac{5}{11}$.
Alors la condition de descente suffisante (4.24) est encore une fois vérifiée.

Démonstration 7. .
Nous essayons de simplifier β_k^{DL} en fonction des conditions que nous avons.

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - s_k)}{d_k^T y_k} = \frac{g_{k+1}^T (g_{k+1} - g_k) - g_{k+1}^T s_k}{d_k^T y_k} = \frac{\| g_{k+1} \|^2 - g_{k+1}^T g_k - g_{k+1}^T s_k}{d_k^T y_k}$$

Puisque $g_{k+1}^T s_k + g_k^T g_{k+1} = 0$ i.e. $g_k^T g_{k+1} = -g_{k+1}^T s_k$, alors

$$\beta_k^{DL} = \frac{\| g_{k+1} \|^2 + g_{k+1}^T s_k - g_{k+1}^T s_k}{d_k^T y_k} = \frac{\| g_{k+1} \|^2}{d_k^T y_k} = \beta_k^{DY}.$$

En multipliant la direction donnée par (4.13) et (4.14) par g_{k+1} et en posant $\beta_k^{DL} = \beta_k^{DY}$, nous obtenons

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + (\varphi_k \beta_k^{DY} + \lambda_k \beta_k^{HS} + (1 - \varphi_k - \lambda_k)\beta_k^{DY}) g_{k+1}^T d_k \quad (4.58)$$

en simplifiant les calcules, nous obtenons

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + (\varphi_k \beta_k^{DY} + \lambda_k \beta_k^{HS} + (1 - \lambda_k)\beta_k^{DY} - \varphi_k \beta_k^{DY}) g_{k+1}^T d_k \quad (4.59)$$

donc

$$g_{k+1}^T d_{k+1} = - \| g_{k+1} \|^2 + (\lambda_k \beta_k^{HS} + (1-\lambda_k)\beta_k^{DY}) g_{k+1}^T d_k, \tag{4.60}$$

en remplaçant les formules de β_k^{DY}, β_k^{HS} *et* β_k^{DL} *dans la relation (4.60), nous obtenons*

$$g_{k+1}^T d_{k+1} = - \| g_{k+1} \|^2 + (\lambda_k \frac{g_{k+1}^T y_k}{d_k^T y_k} + (1-\lambda_k) \frac{\| g_{k+1}^T \|^2}{d_k^T y_k}) g_{k+1}^T d_k \tag{4.61}$$

ce qui donne

$$g_{k+1}^T d_{k+1} = - \| g_{k+1} \|^2 + (\lambda_k \| g_{k+1}^T \|^2 - \lambda_k g_{k+1}^T g_k + \| g_{k+1}^T \|^2 - \lambda_k \| g_{k+1}^T \|^2) \frac{g_{k+1}^T d_k}{d_k^T y_k}$$

ainsi

$$g_{k+1}^T d_{k+1} \leq - \| g_{k+1} \|^2 + | - \lambda_k g_{k+1}^T g_k + \| g_{k+1}^T \|^2 | \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \tag{4.62}$$

alors

$$g_{k+1}^T d_{k+1} \leq - \| g_{k+1} \|^2 + (| - \lambda_k | |g_{k+1}^T g_k| + \| g_{k+1} \|^2) \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|}$$

en utilisant (4.23), il en résulte

$$g_{k+1}^T d_{k+1} \leq - \| g_{k+1} \|^2 + (0.2|\lambda_k| \| g_{k+1} \|^2 + \| g_{k+1} \|^2) \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \tag{4.63}$$

par conséquent

$$g_{k+1}^T d_{k+1} \leq - \| g_{k+1} \|^2 + \frac{(0.2|\lambda_k|+1)\sigma}{1-\sigma} \| g_{k+1} \|^2$$

ce qui donne alors

$$g_{k+1}^T d_{k+1} \leq -(1 - \frac{(0.2|\lambda_k|+1)\sigma}{1-\sigma}) \| g_{k+1} \|^2 .$$

Pour que la condition de descente suffisante soit satisfaite, il suffit que

$$0 < \frac{(0.2|\lambda_k|+1)\sigma}{1-\sigma} < 1. \tag{4.64}$$

En prenant $0 < \sigma < \frac{5}{11}$, *quelle est la valeur de* λ_k *pour atteindre (4.64) ?*
Clairement, nous avons toujours

$$\frac{(0.2|\lambda_k|+1)\sigma}{1-\sigma} > 0, \tag{4.65}$$

car $0 < \sigma < \frac{5}{11}$. *Ensuite, nous allons choisir* λ_k *de telle manière que*

$$\frac{(0.2|\lambda_k|+1)\sigma}{1-\sigma} < 1$$

alors

$$0.2\sigma|\lambda_k| + \sigma < 1 - \sigma$$

ainsi

$$|\lambda_k| < \frac{1-2\sigma}{0.2\sigma}$$

ce qui donne

$$-\frac{1-2\sigma}{0.2\sigma} < \lambda_k < \frac{1-2\sigma}{0.2\sigma}.$$

Il suffit de prendre $\lambda_k = \frac{\frac{1-2\sigma}{0.2\sigma}}{2} = \frac{1-2\sigma}{0.4\sigma}$ ou $\lambda_k = -\frac{\frac{1-2\sigma}{0.2\sigma}}{2} = -\frac{1-2\sigma}{0.4\sigma}$.
Maintenant, nous résumons l'analyse précédente en donnant le couple (λ_k, φ_k) comme suit

$$(\lambda_k, \varphi_k) = \begin{cases} (|g_{k+1}^T s_k + g_k^T g_{k+1}|, -\frac{\lambda_k g_{k+1}^T s_k}{g_{k+1}^T s_k + g_k^T g_{k+1}}) & \text{If} \quad g_{k+1}^T s_k + g_k^T g_{k+1} \neq 0, \\ (\frac{1-2\sigma}{0.4\sigma}, \frac{\lambda_k g_{k+1}^T s_k}{2\sigma g_k^T s_k}) & \text{Sinon} \end{cases} \tag{4.66}$$

où, $0 < \sigma < \frac{5}{11}$.

4.3 Les propriétés de convergence de notre méthode

Les hypothèses essentielles (voir [7]) suivantes sur la fonction objective, sont nécessaires pour établir la convergence globale de notre méthode hybride (QCC).

H1. L'ensemble de niveaux $\mathcal{H} = \{x \in R^n / f(x) \leq f(x_0)\}$ est borné, où x_0 est le vecteur initial.

H2. Dans un certain voisinage $\mathcal{Q}$ de $\mathcal{H}$, la fonction objective f, est continument différentiable, son gradient est continu Lipschitz, par conséquent $\exists l > 0$ tel que

$$\| g(x) - g(y) \| \leq l \| x - y \| \qquad \forall \quad x, y \in \mathcal{Q}. \tag{4.67}$$

Ces hypothèses impliquent que $\exists \eta \geq 0$ tel que

$$\| g(x) \| \leq \eta, \quad \forall x \in \mathcal{H} \tag{4.68}$$

Le lemme 5 de Zoutendijk [60] et le Lemme 6 vont être utilisé pour démontrer le théorème de convergence ci après.

Lemme 5. .
Supposons que les hypothèses H1 et H2 ci-dessus, soient vérifiées.
Considérons toute itération de la forme (4.2), où d_k satisfait à la condition de descente $g_k^T d_k < 0$ et α_k satisfait à la recherche linéaire inexacte de Wolfe, ou à sa version forte,
alors on a

$$\sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\| d_k \|^2} < +\infty \tag{4.69}$$

Lemme 6. *[39].*
Supposons que les hypothèses H1 et H2 ci-dessus soient vérifiées. Si d_k est une direction de descente et que la longueur du pas α_k satisfait

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \quad \sigma < 1, \tag{4.70}$$

alors

$$\alpha_k \geq \frac{1 - \sigma}{l} \frac{|d_k^T g_k|}{\| d_k \|^2}. \tag{4.71}$$

À partir des hypothèses H1, H2 et (4.24), il en résulte que α_k, obtenu dans l'algorithme QCC n'est pas égal à zéro, c'est-à-dire $\exists p > 0$ tel que

$$\alpha_k \geq p, \quad \forall k \geq 0. \tag{4.72}$$

Le théorème suivant donne la convergence globale.

Théoreme 17. .
Supposons que les hypothèses H1 et H2 soient satisfaites, pour tout $x_0 \in \mathbb{R}^n$, soit x_k la séquence générée par l'algorithme QCC.
Alors l'algorithme QCC est globalement convergent, i.e.

$$\lim_{k \mapsto \infty} inf \| g_k \| = 0. \tag{4.73}$$

Démonstration 8. .
la démonstration se fait par un raisonnement par l'absurde.
Supposons que (4.73) soit faux. Alors $\exists r > 0$ tel que

$$\| g_k \| > r. \tag{4.74}$$

En utilisant la deuxième condition de la règle de Wolfe forte inexacte (4.4) et la condition de descente suffisante (4.24), il en résulte que

$$d_k^T y_k \geq -(1 - \sigma) g_k^T d_k \geq (1 - \sigma) c \| g_k \|^2 .$$

D'après (4.74) nous obtenons:

$$d_k^T y_k \geq (1 - \sigma) c r^2,$$

donc

$$\frac{1}{d_k^T y_k} \leq \frac{1}{(1-\sigma)cr^2}. \tag{4.75}$$

D'autre part, ee (4.24) et (4.74), nous avons

$$-g_k^T d_k \geq c \parallel g_k \parallel^2 \geq cr^2 \tag{4.76}$$

ainsi

$$\frac{1}{-g_k^T d_k} \leq \frac{1}{cr^2}. \tag{4.77}$$

Sachant que $y_k = g_{k+1} - g_k$ et $s_k = x_{k+1} - x_k$, selon (4.67), nous avons

$$\parallel y_k \parallel \leq l \parallel s_k \parallel \leq lD, \tag{4.78}$$

où, $D = \max \parallel x - y \parallel / x, y \in \mathcal{H}$ est le diamètre de l'ensemble de niveau $\mathcal{H}$.
D'après la formule de d_{k+1} dans (4.13), nous avons

$$\parallel d_{k+1} \parallel \leq \parallel g_{k+1} \parallel + |\beta_k^{QCC}| \parallel d_k \parallel . \tag{4.79}$$

Sachant que, d'après la formule de β_k^{QCC} donné par (4.14), nous obtenons

$$|\beta_k^{QCC}| = |\varphi_k||\beta_k^{DY}| + |\lambda_k||\beta_k^{HS}| + |(1-\varphi_k-\lambda_k)||\beta_k^{DL}| \tag{4.80}$$

où, d'après les formules de β_k^{DL}, β_k^{DY} et β_k^{HS}, nous avons

$$|\beta_k^{DL}| = \frac{|g_{k+1}^T(y_k - s_k)|}{|d_k^T y_k|} \leq \frac{|g_{k+1}^T y_k| + |g_{k+1}^T s_k)|}{|d_k^T y_k|} \leq \frac{\parallel g_{k+1} \parallel \parallel y_k \parallel + \parallel g_{k+1} \parallel \parallel s_k \parallel}{|d_k^T y_k|}$$

de même que

$$|\beta_k^{DY}| \quad = \quad \frac{\parallel g_{k+1} \parallel^2}{|d_k^T y_k|}$$

et enfin

$$|\beta_k^{HS}| = \frac{|g_{k+1}^T y_k|}{|d_k^T y_k|} \leq \frac{\parallel g_{k+1} \parallel \parallel y_k \parallel}{|d_k^T y_k|}$$

D'après (4.68), (4.75) et (4.78) alors

$$|\beta_k^{DL}| \leq \frac{\eta l D + \eta D}{(1-\sigma)Cr^2} \tag{4.81}$$

ainsi que

$$|\beta_k^{DY}| \leq \frac{\eta^2}{(1-\sigma)Cr^2} \tag{4.82}$$

de plus

$$|\beta_k^{HS}| \leq \frac{\eta l D}{(1-\sigma)Cr^2}. \tag{4.83}$$

Ensuite, d'après (4.81), (4.82) et (4.83) alors la relation (4.80), devient

$$|\beta_k^{QCC}| \quad \leq \quad \frac{|\varphi_k|\eta^2}{(1-\sigma)cr^2} + \frac{|\lambda_k|\eta l D}{(1-\sigma)cr^2} + \frac{|1-\varphi_k-\lambda_k|(\eta l D + \eta D)}{(1-\sigma)cr^2} \tag{4.84}$$

En ce qui concerne la bornitude des paramètres d'hybridation (λ_k, φ_k), il y a deux cas.

1. $(\lambda_k, \varphi_k) = (|g_{k+1}^T s_k + g_k^T g_{k+1}|, -\frac{|g_{k+1}^T s_k + g_k^T g_{k+1}| g_{k+1}^T s_k}{g_{k+1}^T s_k + g_k^T g_{k+1}})$
   D'après (4.23), (4.68) et (4.78)
   $$|\lambda_k| = |g_{k+1}^T s_k + g_k^T g_{k+1}| \leq \parallel g_{k+1} \parallel \parallel s_k \parallel + 0.2 \parallel g_{k+1} \parallel^2 | \leq \eta D + 0.2\eta^2 = a_1$$
   De (4.68) et (4.78)
   $$|\varphi_k| = \frac{|g_{k+1}^T s_k + g_k^T g_{k+1}||g_{k+1}^T s_k|}{|g_{k+1}^T s_k + g_k^T g_{k+1}|} = |g_{k+1}^T s_k| \leq \parallel g_{k+1} \parallel \parallel s_k \parallel \leq \eta D = a_2$$
   $$|1-\varphi_k-\lambda_k| \leq 1 + |\varphi_k| + |\lambda_k| \leq 1 + a_2 + a_1 = a_3$$

2. $(\lambda_k, \varphi_k) = (\frac{1-2\sigma}{0.4\sigma}, \frac{\lambda_k g_{k+1}^T s_k}{2\sigma g_k^T s_k})$

$$|\lambda_k| = |\frac{1-2\sigma}{0.4\sigma}| = \frac{1-2\sigma}{0.4\sigma} = b_1$$

car $0 < \sigma < \frac{5}{11}$
d'après (4.77), (4.68) et (4.78)

$$|\varphi_k| = \frac{|\lambda_k g_{k+1}^T s_k|}{|2\sigma g_k^T s_k|} = \frac{b_1 |g_{k+1}^T s_k|}{2\sigma |g_k^T s_k|} \leq \frac{b_1 \parallel g_{k+1} \parallel \parallel s_k \parallel}{2\sigma |g_k^T s_k|} \leq \frac{b_1 \eta D}{2\sigma c r^2} = b_2$$

$$|1 - \varphi_k - \lambda_k| \leq 1 + |\varphi_k| + |\lambda_k| \leq 1 + b_2 + b_1 = b_3$$

Dans les deux cas, les paramètres sont bornés, donc nous pouvons résumer en disant que

$$|\lambda_k| \leq A \tag{4.85}$$

$$|\varphi_k| \leq B \tag{4.86}$$

$$|1 - \varphi_k - \lambda_k| \quad \leq \quad C \tag{4.87}$$

Ensuite, selon (4.85), (4.86) et (4.87), la relation (4.84) s'écris

$$|\beta_k^{QCC}| \quad \leq \quad \frac{B\eta^2}{(1-\sigma)cr^2} + \frac{A\eta l D}{(1-\sigma)cr^2} + \frac{C(\eta l D + \eta D)}{(1-\sigma)cr^2} = E \tag{4.88}$$

Sachant que $s_k = \alpha_k d_k$, *donc*

$$\parallel d_k \parallel = \frac{\parallel s_k \parallel}{\alpha_k} \tag{4.89}$$

ensuite, utilisant ((4.72)) et (4.78) alors (4.89) s'écris

$$\parallel d_k \parallel \leq \frac{D}{P} \tag{4.90}$$

Par conséquent, à partir de (4.68), (4.88) et (4.90), la direction d_{k+1} *est borné, i.e. (4.79) devient*

$$\parallel d_{k+1} \parallel \leq \eta + E\frac{D}{p} = M, \tag{4.91}$$

avec M *est une constante positive.*
Ce qui donne alors

$$\frac{1}{\parallel d_{k+1} \parallel} \geq \frac{1}{M}.$$

Donc

$$\sum_{k\geq 0} \frac{1}{\parallel d_{k+1} \parallel} \geq \sum_{k\geq 0} \frac{1}{M}.$$

Par conséquent,

$$\sum_{k\geq 0} \frac{1}{\parallel d_k \parallel^2} = +\infty. \tag{4.92}$$

De plus, à partir de (4.69), (4.74) et (4.24), il en résulte que

$$c^2 r^4 \sum_{k\geq 0} \frac{1}{\parallel d_k \parallel^2} \leq \sum_{k\geq 0} \frac{c^2 \parallel g_k \parallel^4}{\parallel d_k \parallel^2} \leq \sum_{k\geq 0} \frac{(g_k^T d_k)^2}{\parallel d_k \parallel^2} < +\infty, \tag{4.93}$$

alors

$$\sum_{k\geq 0} \frac{1}{\parallel d_k \parallel^2} < +\infty.$$

Ceci est une contradiction avec (4.92), donc (4.73) est prouvé.

4.4 Les expériences numériques

Dans cette section, nous analysons les performances de notre algorithme QCC appliqué sur plusieurs problèmes de test d'optimisation sans contrainte choisis de [8]. Chaque problème est testé pour des dimensions allant de 2 à 10000. Nous illustrons l'efficacité de notre algorithme QCC, en le comparant aux les algorithmes DY, HS et DL à partir desquels il a été construit. Nous le comparons également à l'algorithme BFGS, pour confirmer que notre algorithme QCC est une meilleure approximation de la méthode de Newton, que l'algorithme BFGS.

Tous les algorithmes utilisent les conditions de la règle de Wolfe forte, avec les paramètres $\delta = 0.0001$ et $\sigma = 0.1$, ainsi que les mêmes critères d'arrêt $\| g_k \|_\infty \leq 10^{-7}$.
Toutes les expériences numériques sont réalisées sur un PC avec les spécifications suivantes: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50GHz, 4.00 Go de RAM.

Les figures 4.1, 4.2 et 4.3, présentent les performances de ces algorithmes en fonction du temps CPU, du nombre d'itérations et des évaluations du gradient, en utilisant le profil introduit par Dolan et Moré comme outil d'évaluation [22]. Comme toutes les figures ci-dessus le montrent, le nouvel algorithme

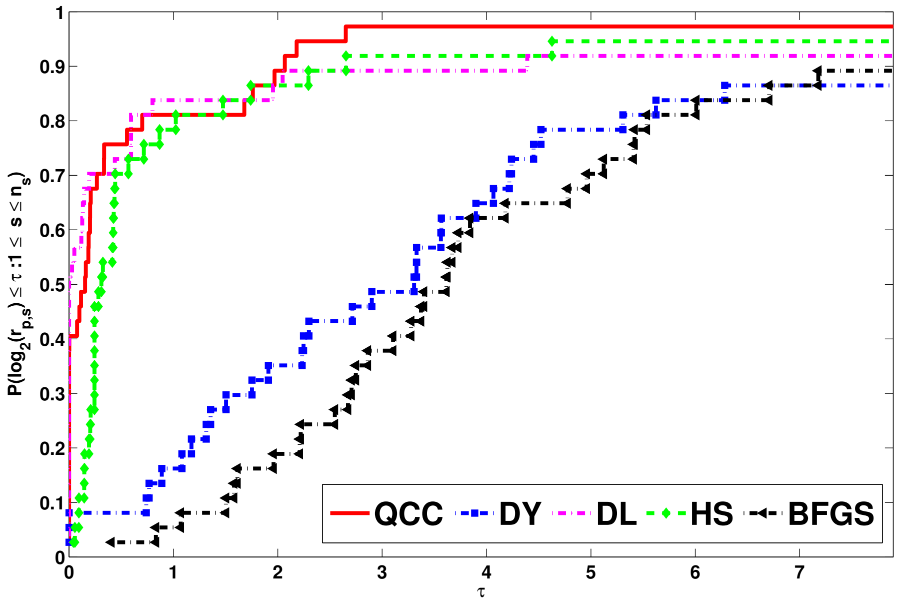


Figure 4.1: Profil de performance pour le temps CPU.

QCC est clairement supérieur aux autres algorithmes.

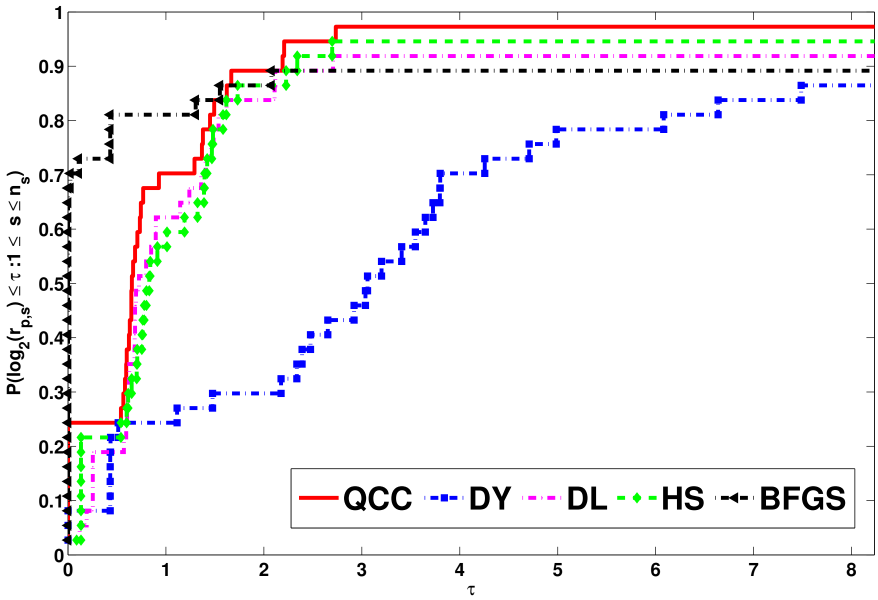


Figure 4.2: Profil de performance pour le nombre d'itérations.

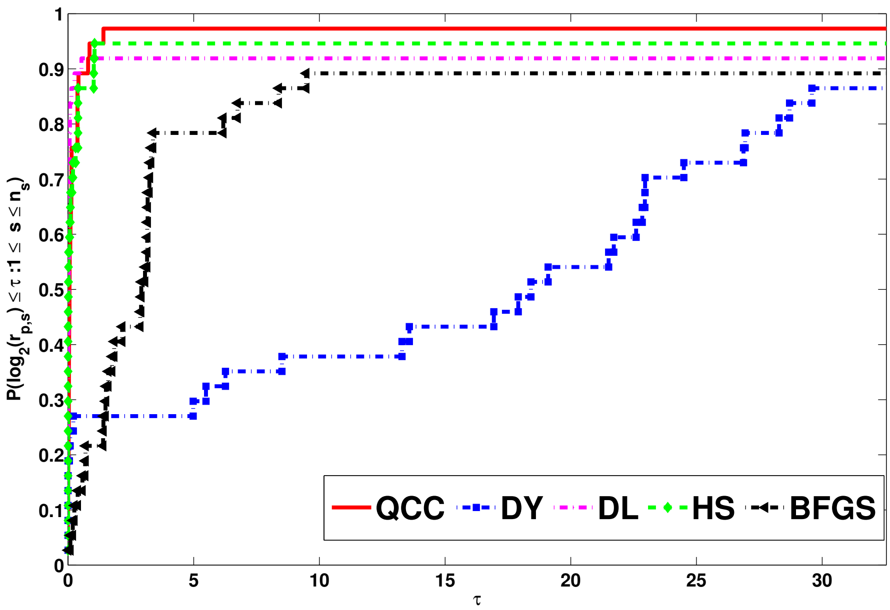


Figure 4.3: Profil de performance pour les évaluations du gradient.

Chapitre 5

Étude numérique de la convergence quadratique

Introduction

La méthode de Newton présente une convergence quadratique, ce qui est très favorable car cela signifie une convergence rapide vers la solution optimale. De point de vue mathématique, au voisinage de la solution optimale, les erreurs $e_k = \| x_k - x^* \|$ (où, x^* est la solution optimale) satisfont l'inégalité,

$$e_{k+1} \leq M e_k^2, \quad \text{pour certains } M \text{ positive.} \tag{5.1}$$

Rappelons que, dans chaque un des chapitres 2 et 3, nous avons développé une nouvelle méthode hybride de gradient conjugué en accord avec la méthode de Newton, nommés (DYDL) et (WYLCD) respectivement, dans le but d'atteindre numériquement le comportement de convergence quadratique observé dans la méthode de Newton, tout en conservant certaines caractéristiques essentielles de l'algorithme de gradient conjugué (expliqué dans les chapitres 2 et 3). De plus, au chapitre 4, nous avons développé une combinaison quasi convexe entre trois directions de descente de l'algorithme du gradient conjugué; de sorte que la nouvelle direction soit construite pour approximer la direction de Newton en utilisant l'équation sécante, cette nouvelle algorithme est nommé (QCC).

Dans ce chapitre, nous présentons un nouveau test, que nous avons développé pour étudier la convergence quadratique de nos trois méthodes hybrides de gradient conjugué. Le but de ce test est d'évaluer si l'algorithme proposé approche numériquement la convergence quadratique en analysant les rapports d'erreur entre deux itérations successives, et montrant graphiquement comment ces rapports d'erreur indiquent la présence d'un convergence quadratique dans le sens numérique.

Ce test constitue une contribution importante de notre travail et offre une méthode fiable pour évaluer l'efficacité de notre méthodes de gradient conjugué.

5.1 Description du test

Le nouveau test que nous avons développé repose sur les étapes suivantes

- Choisissons une fonction de test appropriée de [8].
- Exécutons l'algorithme d'optimisation, on se réfferons à l'algorithme (DYDL) et l'algorithme (WYLCD) ou bien l'algorithme (QCC) dans chapitre 2, 3 et 4 respectivement
- Pour chaque itération, nous calculons les rapport d'erreur r_k avec deux itérations successives,

  $$r_k = \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|^2} \tag{5.2}$$

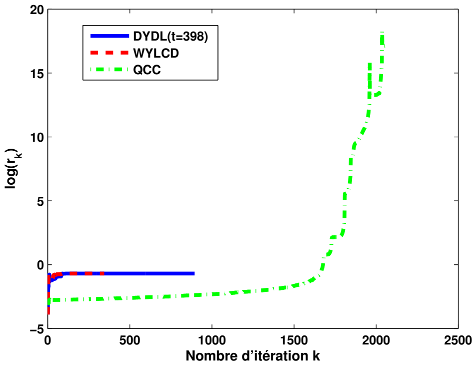
  où, x^* est la solution exacte du problème.

- Nous traçons ensuite le logarithme de ces rapport d'erreur $log(r_k)$ en fonction du nombre d'itérations k pour observer leur comportement.
- Nous nous attendons à ce que ces rapport d'erreur r_k soient majorés ($r_k \leq \max_{k\geq 0}(r_k)$), c'est-à-dire que r_k ne dépassent pas une valeur constante, ce qui serait une indication de la convergence quadratique aux sens numérique de nos méthodes.

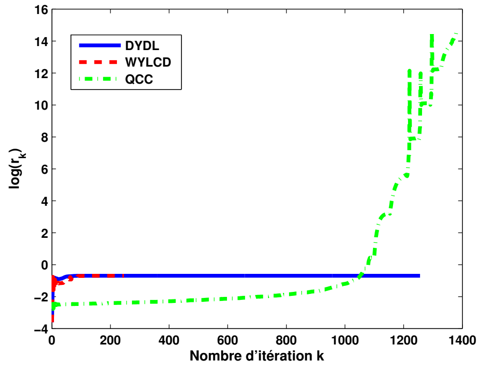
5.2 Exécution du teste

Nous allons étudier la convergence quadratique de notre trois algorithmes (DYDL, WYLCD et QCC) développés respectivement dans les chapitres 2, 3 et 4 en exécutant le teste que nous avons décrits. Nous utiliserons certaines problèmes de test choisies de [8], avec différentes dimensions. Tous les algorithmes appliquent les conditions de recherche linéaire de Wolfe forte; nous choisirons donc les valeurs des paramètres $\delta = 0.0001$ et $\sigma = 0.1$. Nous considérerons également $\| g_k \|_\infty \leq 10^{-6}$ comme critère d'arrêt . Dans les expériences numériques, tous les codes seront exécutés sur un PC ayant les spécifications suivantes: Intel(R) Core(TM) i5-3210M CPU 2.50GHz 2.50 GHz, 4.00 Go RAM.

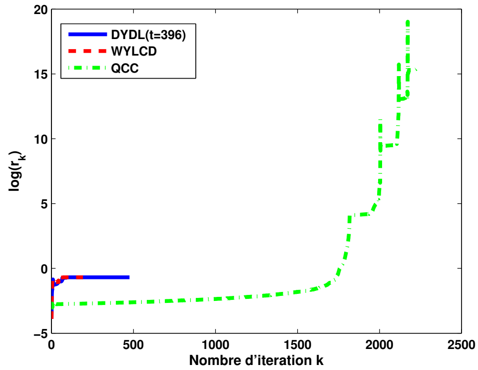
5.3 discussion



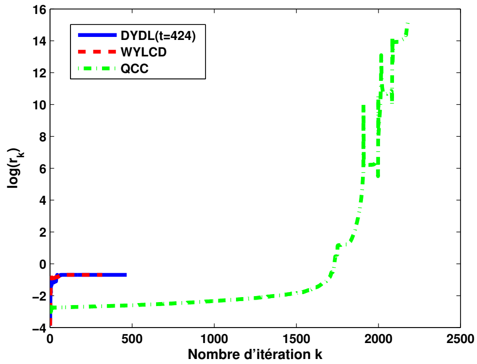
(a) $n = 260$



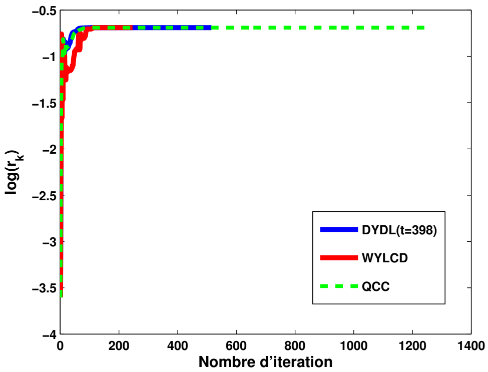
(b) $n = 150$



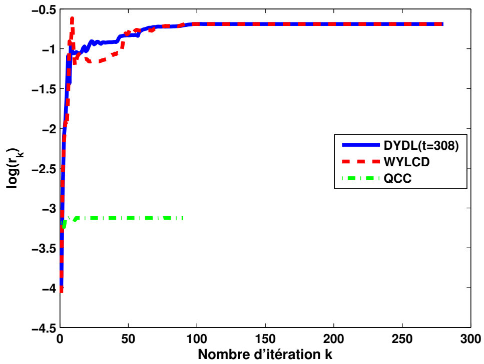
(c) $n = 258$



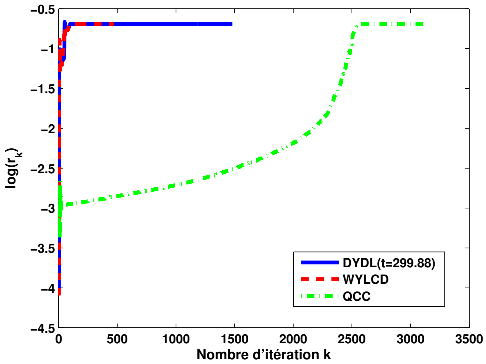
(d) $n = 250$



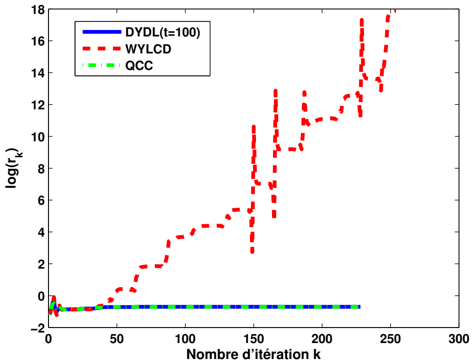
(e) $n = 1005$



(f) $n = 360$



(g) $n = 380$



(h) $n = 5$

Figure 5.1: La convergence quadratique de nos méthodes

Les graphes dans la figure 5.1, illustrent la convergence quadratique de nos trois méthodes (DYDL, WYLCD et QCC).

Lorsque nous analysons des cas tels que $n = 260, n = 150, n = 250$ et $n = 258$, nous observons que les méthodes DYDL et WYLCD convergent en un petit nombre d'itérations, tandis que la méthode QCC nécessite un plus grand nombre d'itérations. De plus, pour les même cas $n = 260, n = 150, n = 250, n = 258$, nous observons que $log(r_k)$ associées aux méthodes (DYDL et WYLCD) ne dépassent pas 0, i.e. $(r_k \leq 1)$; par conséquent, les rapport d'erreur r_k montrent une tendance d'être majoré. Cela indique que les méthodes (DYDL et WYLCD) peuvent avoir un convergence quadratique. Par contre, les r_k pour la méthode (QCC) ne montrent pas de tendance à être majorés. Donc on ne peut pas dire que (QCC) présente une tendance d'avoir un convergence quadratique.

Lorsque nous analysons d'autres cas, tels que $n = 360, n = 380$ et $n = 1005$, nous constatons que les rapport d'erreur r_k associées aux trois méthodes (DYDL) (WYLCD) et (QCC) ne dépassent pas 1, donc apparemment r_k présentent une tendance d'avoir un maximum. Cela indique que dans ces cas, les méthode (DYDL) (WYLCD) et (QCC) présentent également un convergence quadratique.

Ce que signifie que, le développement d'un nouvel algorithme de gradient conjugué en se basant sur la méthode de Newton, marque une avancée significative pour améliorer l'accélération de la convergence dans les problèmes d'optimisation.

Conclusion générale

Notre travail vise à étudier les méthodes d'optimisation sans contraintes et à améliorer leur accélération selon un procède comparative. En particulier, nous nous concentrons sur l'étude des méthodes du gradient conjugué non linéaires et de Newton, dans le but de comprendre comment ces deux méthodes peuvent s'influencer mutuellement et se compléter pour améliorer les performances globales des méthodes d'optimisation.

Afin de répondre à ce souci, nous avons développé trois algorithmes qui permettent de résoudre les problèmes d'optimisation non linéaire en combinant les avantages des méthodes du gradient conjugué et de Newton,, tout en évitant leurs faiblesses.

En premier temps, nous avons développé un nouvel algorithme hybride de gradient conjugué qui relie les méthodes (DY) et (DL) dans un mélange convexe en se basant sur la direction de Newton. Cette méthode a été conçue pour être étroitement liée à la méthode de Newton, tout en évitant le coût de calcul associé à l'évaluation directe de la matrice hessienne grâce à l'utilisation de l'équation de la sécante. Cela la rend utile pour résoudre des problèmes d'optimisation à grande échelle. La condition de descente suffisante et la convergence globale ont été prouvées.

Ensuite, suivant le même principe de notre première hybridation, nous avons élaboré l'hybridation des deux fameuse méthodes (WYL) et (CD) comme combinaison convexe, en se basant sur la direction de Newton. Cette méthode satisfait la condition de descente suffisante et la convergence globale.

Nous avons développé également, une combinaison quasi convexe entre trois directions de descente de l'algorithme du gradient conjugué; de sorte que la nouvelle direction soit construite pour approximer la direction de Newton, en utilisant l'équation sécante. Nous avons démontré la propriété de descente et la convergence globale de cet algorithme.

Nous avons démontré l'efficacité de notre méthodes grâce à des expériences numériques qui confirment leur supériorité par rapport aux algorithmes à partir desquels elles ont été construites, en termes de temps de calcul, de précision et de nombre d'itérations.

Enfin, les méthodes du gradient conjugué et de Newton offrent des avantages distincts, et Sachant que, les trois algorithmes de gradient conjugué développés (DYDL, WYLCD, QCC) sont étroitement liés à la méthode de Newton. L'objectif était de préserver certaines bonnes propriétés des algorithmes de gradient conjugué tout en incorporant certaines propriétés souhaitables de la méthode de Newton. Nous avons donc développé un nouveau test numérique pour tester si les trois algorithmes proposés approche numériquement la convergence quadratique observé dans la méthode de Newton.

Les résultats obtenus ont démontré que les méthodes, DYDL et WYLCD, présentent un convergence quadratique. En revanche, dans certains cas, la méthode QCC présente également un convergence quadratique. Toutefois, dans d' autres cas, cette méthode n'offre pas de convergence quadratique et perd l'avantage de la convergence rapide observée dans les méthodes DYDL et WYLCD.

Cela indique que généralement dans de nombreux cas, la construction d'un nouvel algorithme de gradient conjugué qui s'accorde avec la méthode de Newton constitue une avancée significative pour améliorer l'accélération de la convergence dans les problèmes d'optimisation.

Par conséquent, Notre test numérique constitue une contribution importante de notre travail et offre une technique fiable pour évaluer l'efficacité des nouvelles méthodes d'optimisation.

En tant que perspectives futures, nous essaierons dans un premier temps de continuer à améliorer les

algorithmes du gradient conjugué et de Newton, et d'étendre leur applicabilité à un plus large éventail de problèmes. De plus, il pourrait être intéressant d'explorer d'autres nouvelles techniques d'hybridation, combinant les directions de gradient conjugué et de Quazi Newton.

Bibliographie

[1] A.B. Abubakar et al, *A Liu-Storey-type conjugate gradient method for unconstrained minimization problem with application in motion control*, Journal of King Saud University - Science., 2022, 34(4), 101923.

[2] A.B. Abubakar, P. Kumam, M. Malik and A.H. Ibrahim, *A hybrid conjugate gradient based approach for solving unconstrained optimization and motion control problems*, Mathematics and Computers in Simulation., 2022, 201, 640-657.

[3] Alhawarat, A., Salleh, D., Masmali, I.A. : A convex combination between two different search directions of conjugate gradient method and application in image restoration. Mathematical Problems in Engineering (2021).

[4] Andrei, Neculai. "Hybrid conjugate gradient algorithm for unconstrained optimization." Journal of Optimization Theory and Applications 141 (2009) : 249-264.

[5] Andrei, N. : (2008) A Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization as a Convex Combination of Hestenes-Stiefel and Dai-Yuan. Studies in Informatics and Control. **17**(1), 57 (2008) .

[6] Andrei, N. : Another hybrid conjugate gradient algorithm for unconstrained optimization. Numer. Algor. **47**(2), 143–156 (2008).

[7] Andrei, N. : A hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan algorithms. Studies in Informatics and Control. **17**(4), 373–392 (2008).

[8] Andrei, N. : An unconstrained optimization test functions. collection. Adv. Model. Optim. **10**(1), 147–161 (2008).

[9] Akinduko, O.B. : A new conjugate gradient method with sufficient descent property. Earthline Journal of Mathematical Sciences. **6**(1), 163–174 (2021).

[10] Armijo, Larry. "Minimization of functions having Lipschitz continuous first partial derivatives." Pacific Journal of mathematics 16.1 (1966) : 1-3.

[11] A. Samson, *Nonlinear Programming : Theories and Algorithms of Some Unconstrained Optimization Methods (Steepest Descent and Newton's Method)*, International Journal of Engineering and Management Research., 2020, 10, 2250-0758.

[12] Bazaraa, Mokhtar S., Hanif D. Sherali, and Chitharanjan M. Shetty. Nonlinear programming : theory and algorithms. John wiley, sons, 2013.

[13] Beck, Amir. Introduction to nonlinear optimization : Theory, algorithms, and applications with MATLAB. Society for Industrial and Applied Mathematics, 2014.

[14] B.T. Polyak, *Newton's method and its use in optimization*, European Journal of Operational Research., 2007, 181, 1086–1096.

[15] Dai, Y.H., Yuan, Y. : A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on optimization. **10**(1), 177–182 (1999).

[16] Dai, Y.H., Liao, L.Z. : New Conjugacy Conditions and Related Nonlinear Conjugate Gradient Methods. Appl. Math. Optim. **43**(1), 87–101 (2001).

[17] Dai, Yu-Hong. "New properties of a nonlinear conjugate gradient method." Numerische Mathematik 89 (2001) : 83-98.

[18] Dai, Y. H., and Yaxiang Yuan. "Convergence properties of the conjugate descent method." Advances in Mathematics 25.6 (1996) : 552-562.

[19] Djordjević, S.S. : New hybrid conjugate gradient method as a convex combination of FR and PRP methods. Filomat. **30**(11), 3083–3100 (2016).

[20] Djordjević, S.S. : New hybrid conjugate gradient method as a convex combination of LS and CD methods. Filomat. **31**(6), 1813–1825 (2017).

[21] Djordjević, S.S. : New Hybrid Conjugate Gradient Method As A Convex Combination of Ls and Fr Methods. Acta. Math. Sci. **39**(1), 214–228 (2019).

[22] Dolan, E.D., Moré, J.J. : Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002).

[23] Fletcher, R., Reeves, C.M. : Function minimization by conjugate gradients. Comput. J. **7**(2), 149–154 (1964).

[24] Fletcher, R. : Practical methods of optimization. Unconstrained Optimization. Vol. 1, John Wiley and Sons, New York (1987).

[25] Goldstein, A. A., and J. F. Price. "An effective algorithm for minimization." Numerische Mathematik 10 (1967) : 184-189.

[26] Hamel, N., Benrabia, N., Ghiat, M., Guebbai, H. : (2023) A new hybrid conjugate gradient algorithm based on the Newton direction to solve unconstrained optimization problems. J. Appl. Math. Comput. **69**, 2531–2548 (2023).

[27] Hamel, N., Benrabia, N., Ghiat, M and Guebbai, H. (2024). DEVELOPING A NEW CONJUGATE GRADIENT ALGORITHM WITH THE BENEFIT OF SOME DESIRABLE PROPERTIES OF THE NEWTON ALGORITHM FOR UNCONSTRAINED OPTIMIZATION. Journal of Applied Analysis and Computation, 14(1), 458-472.

[28] Hanachi, S.B., Sellami, B., Belloufi, M. : (2022) New iterative conjugate gradient method for nonlinear unconstrained optimization. RAIRO-Operations Research. **56**(4), 2315–2327 (2022).

[29] Hallal, A., Belloufi, M., Sellami, B. : An efficient new hybrid CG-method as convex combination of DY and CD and HS algorithms. RAIRO-Operations Research. **56**(6), 4047 ?4056 (2022).

[30] Hassan, B.A., Owaid, O.A., Yasen, Z.T. : A variant of hybrid conjugate gradient methods based on the convex combination for optimization. Indonesian Journal of Electrical Engineering and Computer Science. **20**(2), 1007–1015 (2020).

[31] Hestenes, M.R., Stiefel, E.L. : Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**(6), 409–436 (1952).

[32] Hu, Y. F., and C. Storey. "Global convergence result for conjugate gradient methods." Journal of Optimization Theory and Applications 71.2 (1991) : 399-405.

[33] H. Huang, Z. Wei and Y. Shengwei, *The proof of the sufficient descent condition of the Wei-Yao-Liu conjugate gradient method under the strong Wolfe-Powell line search*, Applied Mathematics and Computation., 2007, 189, 1241–1245.

[34] H.A. Wasi and M.A.K.Shiker, *Nonlinear Conjugate Gradient Method with Modified Armijo Condition to Solve Unconstrained Optimization.* J. Phys. : Conf. Ser., 2021, 1818, 1–7.

[35] H. Oviedo, *Implicit steepest descent algorithm for optimization with orthogonality constraints*, Optimization Letters., 2022, 16(6), 1773-1797.

[36] H.J.M. Shi, Y. Xie, R. Byrd and J. Nocedal. *A noise-tolerant quasi-Newton algorithm for unconstrained optimization*, SIAM Journal on Optimization., 2022, 32(1), 29–55.

[37] Jardow, F.N., Al-Naemi, G.M. : A new hybrid conjugate gradient algorithm as a convex combination of MMWU and RMIL nonlinear problems. Journal of Interdisciplinary Mathematics. **24**(3), 637–655 (2021).

[38] Kaelo, P., Narayanan, S., Thuto, M.V. : A modified quadratic hybridization of Polak-Ribiere-Polyak and Fletcher-Reeves conjugate gradient method for unconstrained optimization problems. An International Journal of Optimization and Control : Theories-Applications (IJOCTA). **7**(2), 177–185 (2017).

[39] Liu, J.K., Li, S.J. : New hybrid conjugate gradient method for unconstrained optimization. Applied Mathematics and Computation. **245**, 36–43 (2014).

[40] Liu, Y., Storey, C. : Efficient generalized conjugate gradient algorithms, part 1 : theory. J. Optim. Theory. Appl. **69**(1), 129–137 (1991).

[41] M. Al-Baali, Descent property and global convergence of the Fletcher Reeves method with inexact line search, IMA J. Numer. Anal. **5**, 121–124 (1985).

[42] Mohamed, N.S., Mamat, M., Rivaie, M., Shaharudin,S.M. : A comparison on classical-hybrid conjugate gradient method under exact line search. International Journal of Advances in Intelligent Informatics. **5**(2), 150–168 (2019).

[43] Narayanan, S., Kaelo, P. : A Linear Hybridization of Dai-Yuan and Hestenes-Stiefel Conjugate Gradient Method for Unconstrained Optimization. Numer. Math. Theor. Meth. Appl. **14**, 527–539 (2021).

[44] Nocedal, Jorge, and Stephen J. Wright, eds. Numerical optimization. New York, NY : Springer New York, 1999.

[45] Polak, E., Ribiére, G. : Note sur la convergence de méthodes de directions conjuguées. Rev. Fr. Inf. Rech. Oper. **3**(16), 35–43 (1969).

[46] Polyak, B.T. : The conjugate gradient method in extremal problems. USSR Comput. Math. Math. Phys. **9**(4), 94–112 (1969).

[47] Powell, M.J.D. : Restart procedures for the conjugate gradient method. Math. Program. **12**, 241 ?254 (1977)

[48] Powell, Michael JD. "Nonconvex minimization calculations and the conjugate gradient method." Numerical Analysis : Proceedings of the 10th Biennial Conference held at Dundee, Scotland, June 28–July 1, 1983. Springer Berlin Heidelberg, 1984.

[49] Sun, Wenyu, and Ya-Xiang Yuan. Optimization theory and methods : nonlinear programming. Vol. 1. Springer Science, Business Media, 2006.

[50] Touati-Ahmed, D., and C. Storey. "Efficient hybrid conjugate gradient techniques." Journal of optimization theory and applications 64 (1990) : 379-397.

[51] Wolfe, P. : Convergence Conditions for Ascent Methods. SIAM review. **11**(2), 226–235 (1969).

[52] Wolfe, P. : Convergence Conditions for Ascent Methods. II : Some corrections. SIAM review. **13**(2), 185–188 (1971).

[53] X.P. Zhao, J.C. Yao and Y. Yao, *A nonmonotone gradient method for constrained multiobjective optimization problems*, J. Nonlinear Var. Anal., 2022, 6(6), 693-706.

[54] Xu, X., Kong, F.Y. : New hybrid conjugate gradient methods with the generalized Wolfe line search. SpringerPlus. **5**(1), 1–10 (2016).

[55] Y. Wang, F. Alpak, G. Gao, C. Chen, J. Vink, T. Wells and F. Saaf, *An efficient bi-objective optimization workflow using the distributed quasi-Newton method and its application to well-location optimization*, SPE Journal., 2022, 21(1), 364–380.

[56] Y. Liu, Z. Zhu and B. Zhang, *Two sufficient descent three-term conjugate gradient methods for unconstrained optimization problems with applications in compressive sensing.* J. Appl. Math. Comput., 2022, 68, 1787–1816.

[57] Yuan*, Ya-Xiang. "Analysis on the conjugate gradient method." Dynamical Systems 2.1 (1993) : 19-29.

[58] Z.M. Abdullah and I.K. Jamalaldeen, *A New Hybrid of DY and CGSD Conjugate Gradient Methods for Solving Unconstrained Optimization Problems*, Tik. J. of Pure Sci., 2021, 26(5), 86-91.

[59] Zhang, J.Z., Deng, N.Y., Chen, L.H. : New Quasi-Newton Equation and Related Methods for Unconstrained Optimization. Journal of Optimization Theory and Applications. **102** 147–167 (2009).

[60] Zoutendijk, G. : Nonlinear programming, computational methods. Integer and nonlinear programming. 37–86 (1970).

[61] Z. Wei, S. Yao and L. Liu, *The convergence properties of some new conjugate gradient methods*, Applied Mathematics and Computation., 2006, 183, 1341–1350.